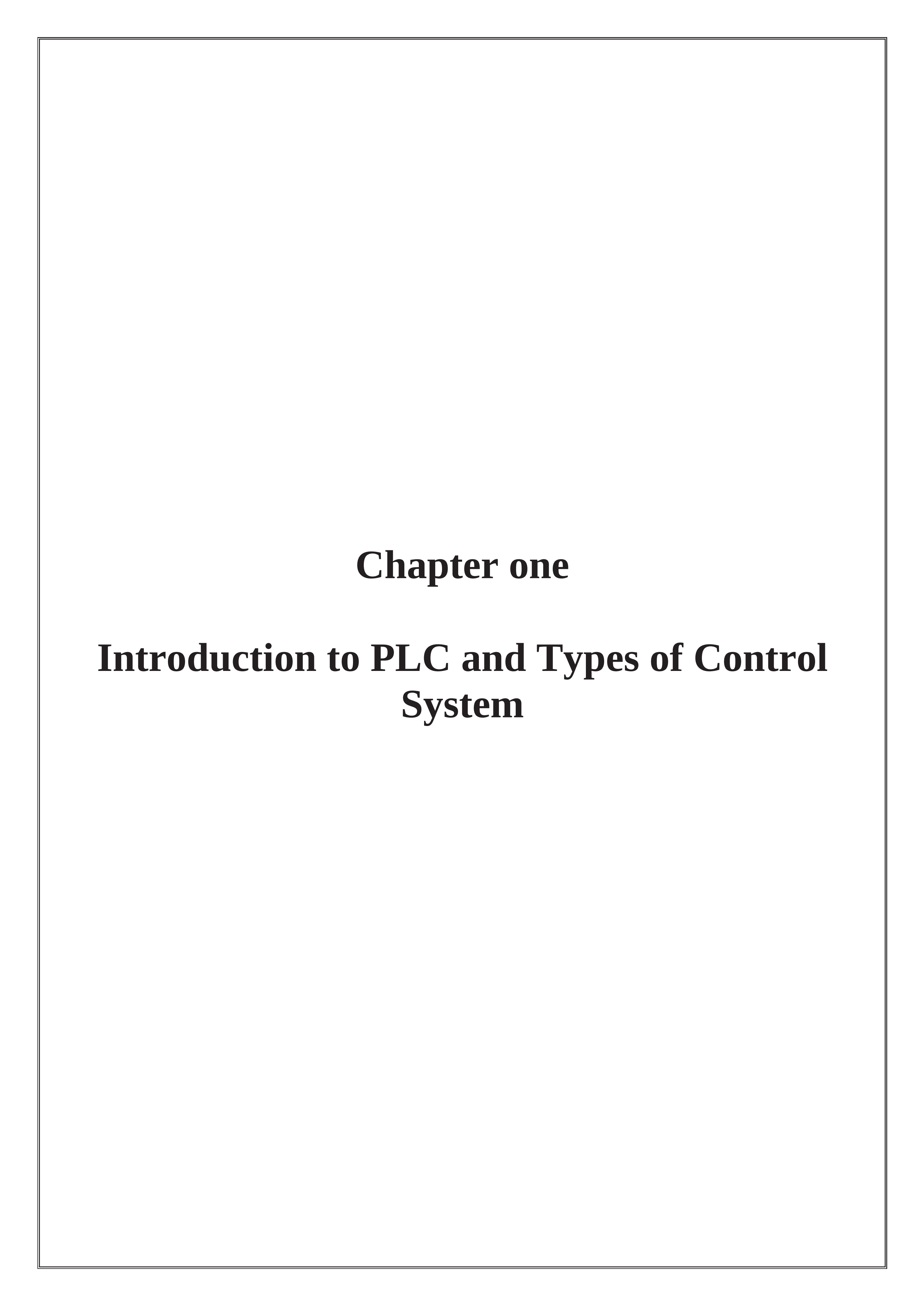

EEE
العربية

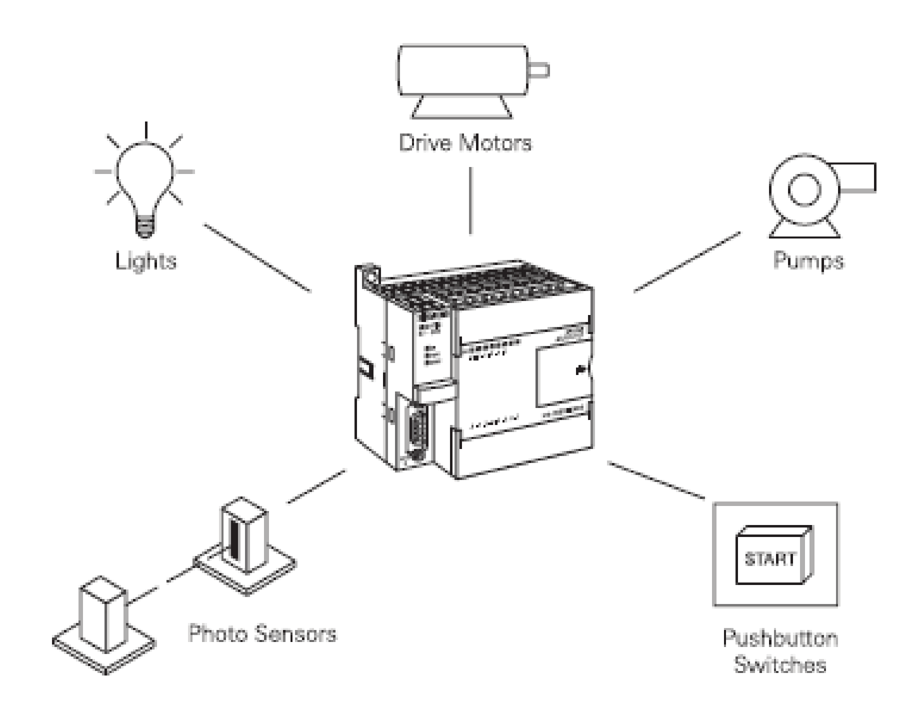


Chapter one

Introduction to PLC and Types of Control System

Introduction to PLC

Programmable Logic Controllers (PLCs), also referred to as programmable controllers, are in the computer family. They are used in commercial and industrial applications. A PLC monitors inputs, makes decisions based on its program, and controls outputs to automate a process or machine. This course is meant to supply you with basic information on the functions and configurations of PLCs .



History of PLC

During the Industrial Revolution of the ١٨th-and ١٩th-centuries, many traditionally manual processes were taken over by machines. These early machines relied on gears and pulleys to work and were, by our standards, extremely primitive. The first major breakthrough in the development of

control systems came with the invention of electrically powered machines. The first control systems were developed in the early years of the ٢٠th century and used sequential Relay Circuits for machine control. A major technical breakthrough in its day, and still used in some plants today, relay technology enabled machines to work faster and more safely.

Relay circuits performed their job very well, but they required large amounts of floor space, and huge amounts of energy. Adding to their drawbacks as the basis for a machine control system, relay circuits also took a long time to install, troubleshoot, and modify. Finally, in the early ١٩٧٠s, a device was developed to replace sequential relay circuits: the Programmable Logic Controller (PLC).

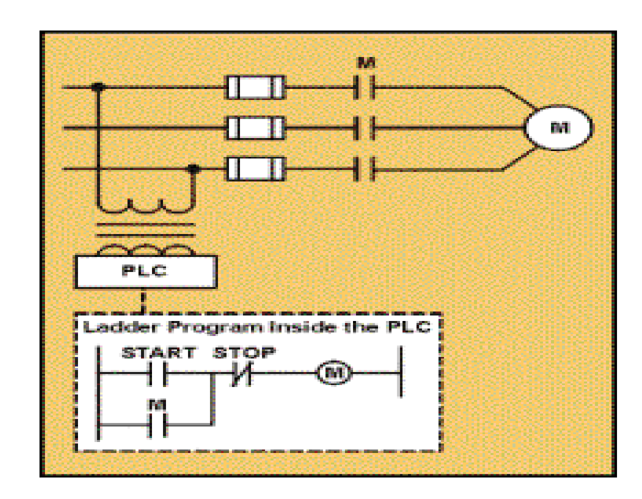
As you will remember from reading about them in Module ٢٤, PLCs are more reliable, faster, more flexible and more efficient than relay-based systems. For example, PLCs are cheaper and easier to wire and maintain than relays. Furthermore, when it comes to troubleshooting, PLCs are much quicker than relays at testing and debugging the program.

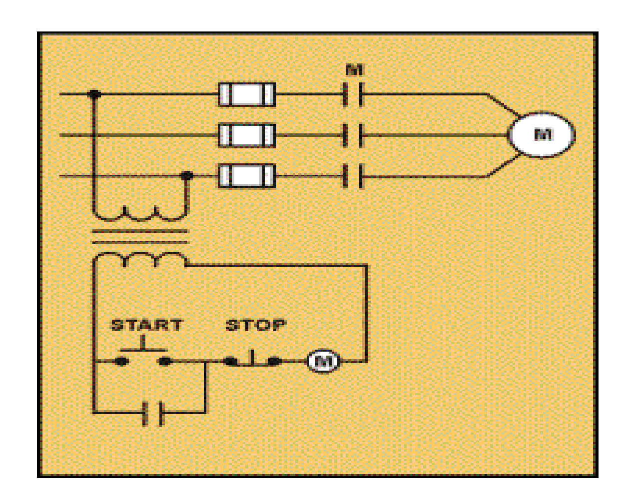
PLCs are used in all kinds of industries. In fact, almost any industrial process that uses electrical control needs a PLC. For example, let's assume that when a switch turns on we want to turn a solenoid on for ٥ seconds and then turn it off regardless of how long the switch is on. We can do this with a simple external timer. But what if the process included ١٠ switches and solenoids?

We would need ١٠ external timers. What if the process also needed to count how many times the switches individually turned on?

We need a lot of external counters. With a PLC, however, we can dispense with those unwieldy timers and counters, and simply program the PLC to count its inputs and turn the solenoids on for the specified time.

The following figure is a Traditional Relay Logic & PLC logic circuit .





- Comparison of PLC with Other Control Systems :-

C\Cs	Relay systems	Digital Logics	Computers	PLC systems
Price Per Function	**Fairly Low**	**Low**	**High**	**Low**
Physical Size	**Bulky**	**Very Compact**	**Fairly Compact**	**Very Compact**
Operating Speed	**Slow**	**Very Fast**	**Fairly Fast**	**Fast**
Noise Immunity	**Excellent**	**Good**	**Fairly Good**	**Good**

Installation	Time Consuming in Design and Install	Time Consuming in Design	Time Consuming in Programming	Easy in Programming and Install
Complex Operation	None	Yes	Yes	Yes
Ease of Changes	Very Difficult	Difficult	Quite Simple	Very Simple
Easy of Maintenance	Poor-large No. Of Contacts	Poor if ICs Soldered	Poor-several Custom Boards	Good-few Standard Cards

Advantages of PLCs : -

The same, as well as more complex tasks, can be done with a PLC. Wiring between devices and relay contacts is done in the PLC program. Hard-wiring, though still required to connect field devices, is less intensive. Modifying the application and correcting errors are easier to handle. It is easier to create and change a program in a PLC than it is to wire and rewire a circuit.

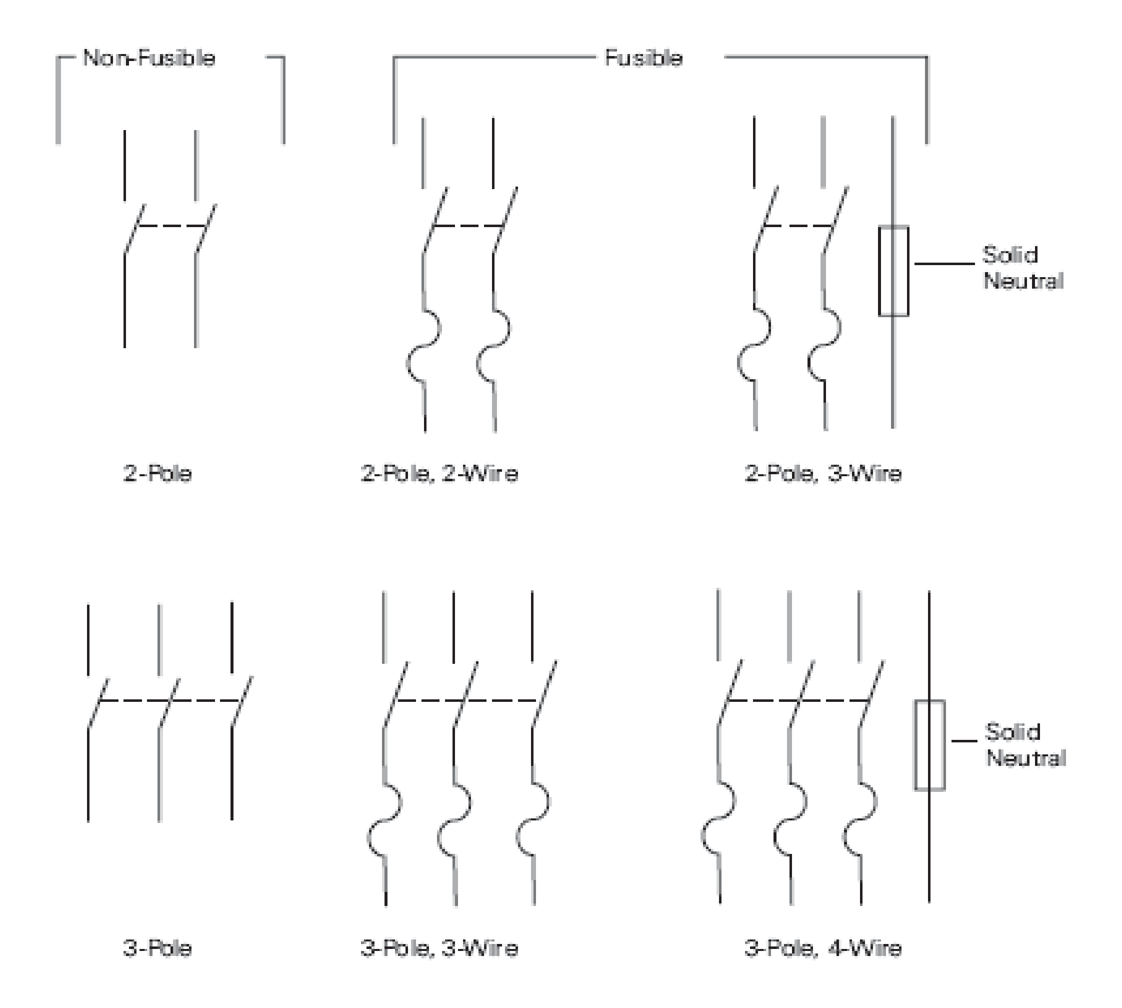
Following are just a few of the advantages of PLCs : -

- Smaller physical size than hard-wire solutions.
- Easier and faster to make changes.
- PLCs have integrated diagnostics and override functions.
- Diagnostics are centrally available.
- Applications can be immediately documented.
- Applications can be duplicated faster and less expensively.

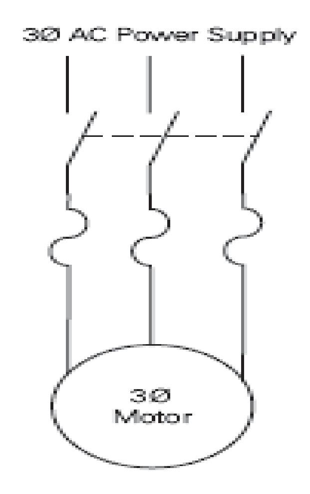
١- Basic Elements of Control System : -

١.١ - Switch Circuit Types : -

The Following diagrams are circuit configuration for ٢- and ٣-pole safety switches. Safety switches may be fusible, non-fusible, or fusible with a solid neutral.



The circuit configuration required depends on the load and on the power supply connected to it. For example, a three-phase motor needs a ٣-pole switch to connect it to a three-phase power supply. If over current protection is required, a fusible ٣-pole safety switch should be selected, as in the following example.



Selecting a Switch : -

There are three important features to consider when selecting a switch:

- **Contacts (e.g. single pole, double throw)**
- **Ratings (maximum voltage and current)**
- **Method of Operation (toggle, slide, key etc.)**

Switch Contacts : -

Several terms are used to describe switch contacts:

- **Pole - number of switch contact sets.**
- **Throw - number of conducting positions, single or double.**
- **Way - number of conducting positions, three or more.**
- **Momentary - switch returns to its normal position when released.**
- **Open - off position, contacts not conducting.**
- **Closed - on position, contacts conducting, there may be several on positions.**

For example: the simplest on-off switch has one set of contacts (single pole) and one switching position which conducts (single throw). The switch mechanism has two positions: open (off) and closed (on), but it is called 'single throw' because only one position conducts.

Switch Contact Ratings : -

Switch contacts are rated with a maximum voltage and current, and there may be different ratings for AC and DC. The AC values are higher because the current falls to zero many times each second and an arc is less likely to form across the switch contacts.

For low voltage electronics projects the voltage rating will not matter, but you may need to check the current rating. The maximum current is less for inductive loads (coils and motors) because they cause more sparking at the contacts when switched off.

Standard Switches : -

Type of Switch | **Circuit Symbol** | **Example**

ON-OFF
Single Pole, Single Throw = SPST

A simple on-off switch. This type can be used to switch the power supply to a circuit.

When used with mains electricity this type of switch *must* be in the live wire, but it is better to use a DPST switch to isolate both live and neutral.

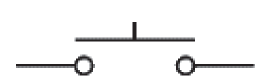




SPST toggle switch

(ON)-OFF
Push-to-make = SPST Momentary

A push-to-make switch returns to its normally open (off) position when you release the button, this is shown by the brackets around ON. This is the standard doorbell switch.





Push-to-make switch

ON-(OFF)

Push-to-break = SPST Momentary

A push-to-break switch returns to its normally closed (on) position when you release the button.



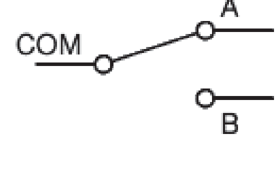


Push-to-break switch

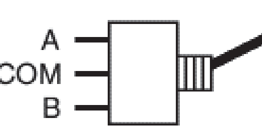
ON-ON
Single Pole, Double Throw = SPDT


SPDT toggle switch

This switch can be on in both positions, switching on a separate device in each case. It is often called a changeover switch. For example, a SPDT switch can be used to switch on a red lamp in one position and a green lamp in the other position.



A SPDT toggle switch may be used as a simple on-off switch by connecting to COM and one of the A or B terminals shown in the diagram. A and B are interchangeable so switches are usually not labeled.




SPDT slide switch (PCB mounting)

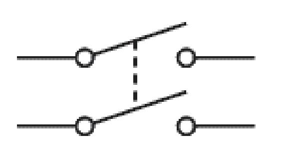
ON-OFF-ON
SPDT Centre Off

A special version of the standard SPDT switch. It has a third switching position in the centre which is off. Momentary (ON)-OFF-(ON) versions are also available where the switch returns to the central off position when released.


SPDT rocker switch

Dual ON-OFF
Double Pole, Single Throw = DPST

A pair of on-off switches which operate together (shown by the dotted line in the circuit symbol).



A DPST switch is often used to switch mains electricity because it can isolate both the live and neutral connections.

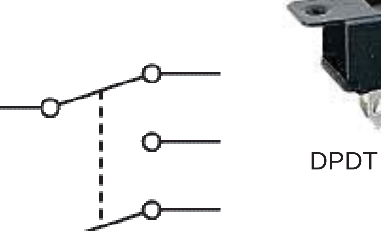

DPST rocker switch

Dual ON-ON

Double Pole, Double Throw = DPDT

A pair of on-on switches which operate together (shown by the dotted line in the circuit symbol).

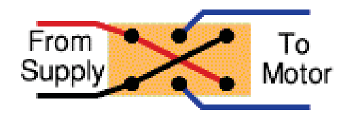
A DPDT switch can be wired up as a reversing switch for a motor as shown in the diagram.




DPDT slide switch

ON-OFF-ON
DPDT Centre Off

A special version of the standard SPDT switch. It has a third switching position in the centre which is off. This can be very useful for motor control because you have forward, off and reverse positions. Momentary (ON)-OFF-(ON) versions are also available where the switch returns to the central off position when released.



Wiring for Reversing Switch

Special Switches : -

Type of Switch **Example**

Push-Push Switch (e.g. SPST = ON-OFF)



This looks like a momentary action push switch but it is a standard on-off switch: push once to switch on, push again to switch off. This is called a latching action.



Micro switch (usually SPDT = ON-ON)



Micro switches are designed to switch fully open or closed in response to small movements. They are available with levers and rollers attached.

Key switch



A key operated switch. The example shown is SPST.

Tilt Switch (SPST)



Tilt switches contain a conductive liquid and when tilted this bridges the contacts inside, closing the switch. They can be used as a sensor to detect the position of an object. Some tilt switches contain mercury which is poisonous.

Reed Switch (usually SPST)



The contacts of a reed switch are closed by bringing a small magnet near the switch. They are used in security circuits, for example to check that doors are closed. Standard reed switches are SPST (simple on-off) but SPDT (changeover) versions are also available.

Warning: reed switches have a glass body which is easily broken!

DIP Switch (DIP = Dual In-line Parallel)



This is a set of miniature SPST on-off switches, the example shown has ٨ switches. The package is the same size as a standard DIL (Dual In-Line) integrated circuit.

This type of switch is used to set up circuits, e.g. setting the code of a remote control.

Multi-pole Switch



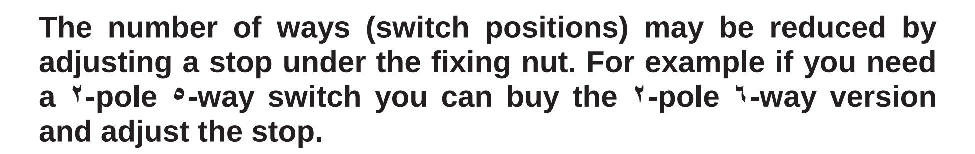
The picture shows a ٦-pole double throw switch, also known as a ٦-pole changeover switch. It can be set to have momentary or latching action. Latching action means it behaves as a push-push switch, push once for the first position, push again for the second position etc.

Multi-way Switch

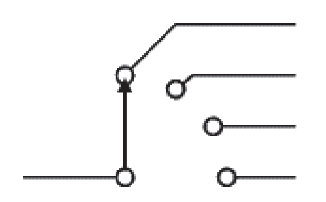


Multi-way rotary switch

Multi-way switches have ٣ or more conducting positions. They may have several poles (contact sets). A popular type has a rotary action and it is available with a range of contact arrangements from ١-pole ١٢-way to ٤-pole ٣ way.



The number of ways (switch positions) may be reduced by adjusting a stop under the fixing nut. For example if you need a ٢-pole ٥-way switch you can buy the ٢-pole ٦-way version and adjust the stop.



١-pole ٤-way switch symbol

Contrast this multi-way switch (many switch positions) with the multi-pole switch (many contact sets) described above.

Fig. ١

Switches are used to open/close a circuit.

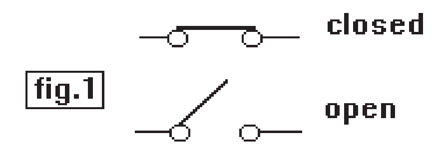
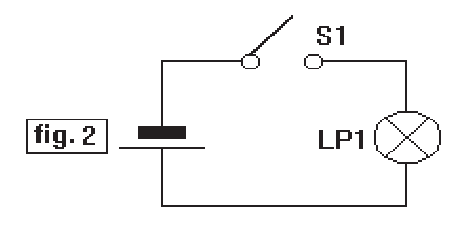


Fig. ٢

S١ is a "single pole on/off" switch in the off position.



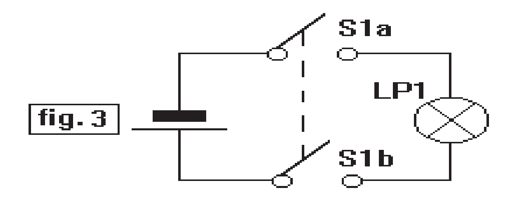


Fig. ٣

This is a "٢ pole on/off" switch which completely isolates the lamp from the supply in the off position.

This may be important if it is a high voltage supply.

The dotted line indicates that S١a and S١b are part of the same switch "ganged" together and operate simultaneously.

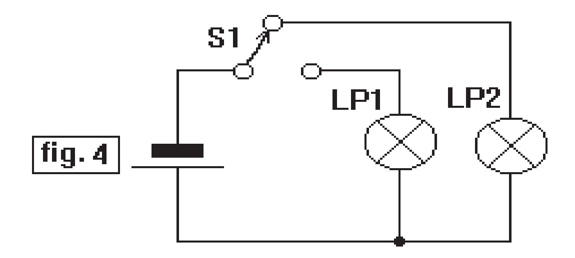


Fig. ٤) : -This is a "single pole changeover" switch. Either lamp ١ or lamp ٢ is on.

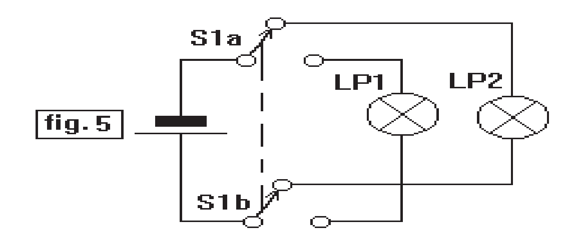


Fig. ٥) :- This is a "٢ pole changeover" switch. The unlit lamp is completely isolated from the supply. Again S١a and S١b are part of the same switch.

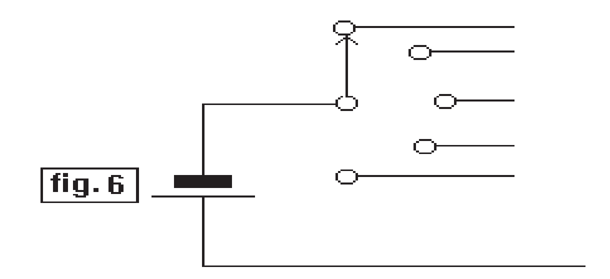


Fig. ٦

This is a "single pole ٥ way" switch. It can select ١ of ٥ circuits. You can have ٢p ٥w, ٣p ٤w etc.

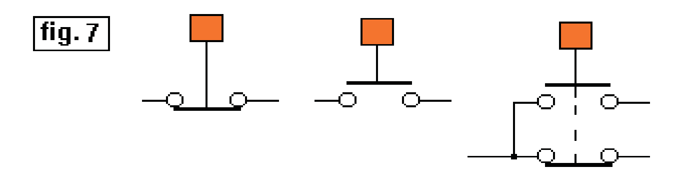


Fig. ٧) This shows :-

(١) a "normally closed, push to break".

(٢) a "normally open, push to make".

(٣) both used together to make a "changeover" switch.

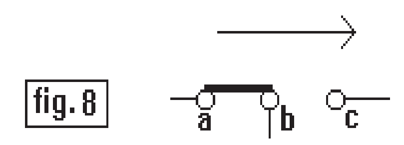


Fig. ٨) : - This is a "changeover" slide switch. When operated a-b opens and b-c closes.

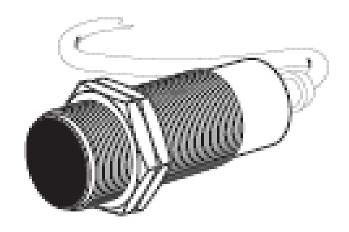
١.٢- Sensors :-

Generally there are ٥ steps to determine which switch type is best suited to the application. This depends on the material properties of the target to be detected.

Step (١) : - type of sensor .
Step (٢) : - Housing design .
Step (٣) : - Sensing range (mm)
Step (٤) : - Electrical data and connections
Step (٥) : - General specifications

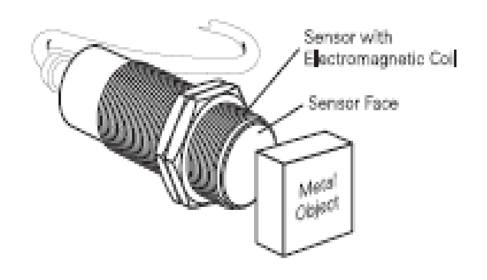
- **Proximity Sensor:**

A type of sensing switch that detects the presence or absence of an object without physical contact.



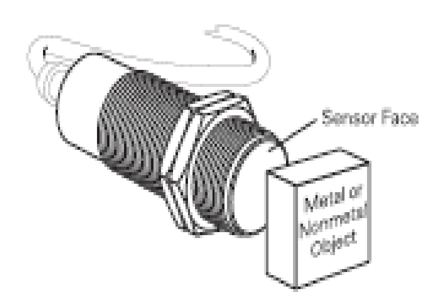
- **Inductive Proximity Sensor:-**

A type of *sensing switch* that uses an electromagnetic coil to detect the presence of a metal object without coming into physical contact with it. Inductive proximity sensors ignore nonmetallic objects.



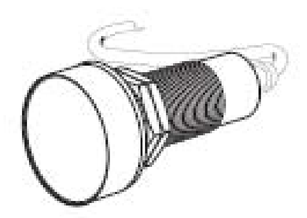
- **Capacitive Proximity Sensor :-**

A type of *sensing switch* that produces an electrostatic field to detect the presence of metal and nonmetallic objects without coming into contact with them.



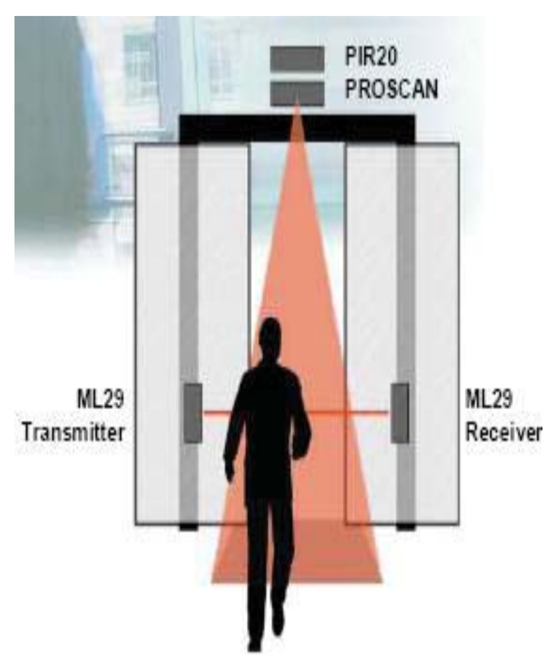
- **Ultrasonic Sensor**

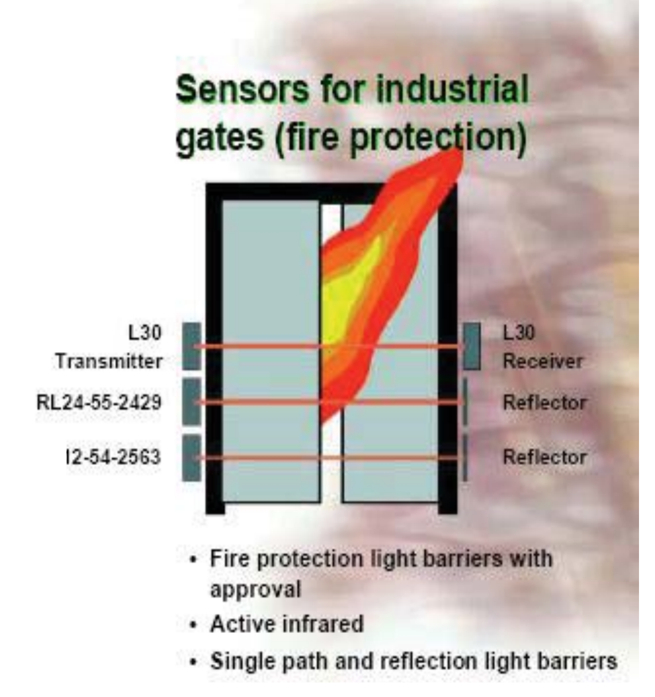
A type of sensing switch that uses high frequency sound to detect the presence of an object without coming into contact with the object.



- **Photoelectric Sensor : -**

Recognition, detection, positioning, classification, counting, notification and monitoring. Nowadays, these processes are largely handled by non-contact photoelectric sensors. Applications range from the automobile industry, mechanical engineering, and assembly automation, through warehousing and conveyor systems and packaging applications, to the printing and paper industries, and naturally include monitoring and safety systems_ .



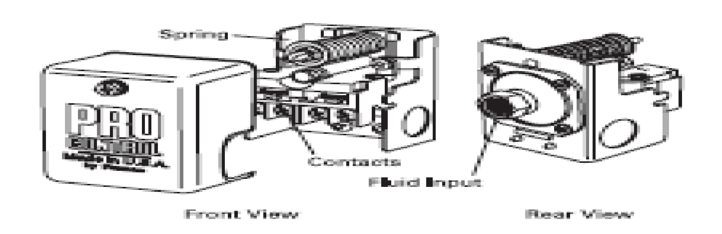






- **<u>Pressure Switch : -</u>**

A control device that opens or closes its contacts in response to a change in the pressure of a liquid or gas.



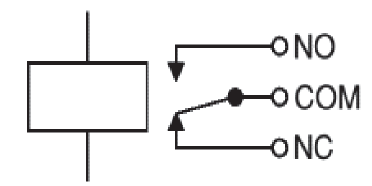
- **Sensing Switches :-**

A device, often called a sensor, used to provide information on the presence or absence of an object. Examples include a limit switch, photoelectric sensor, inductive proximity sensor, capacitive proximity sensor, and ultrasonic proximity sensor.

Sensors	Advantages	Disadvantage	Applications
Limit Switch	• High Current Capability • Low Cost • Familiar " Low-Tech " Sensing	• Require Physical Contact • Very Slow Response • Contact Bounce	• Interlocking • Basic End Travel Sensing
Photoelectric	• Senses all Kinds of Materials • Long Life • Largest Sensing Range • Very Fast Response Time	• Lens Subject to Contamination. • Sensing Range Affected by Color and Reflectivity	• Packaging • Material Handling • Parts Detection
Inductive	• Resistant to Harsh Environments • Very Predictable • Long Life • Easy to Install	• Distance Limitations • Senses Metal Only	• Industrial and Machines. • Machine Tools
Capacitive	• Can Detect Non-Metallic • Detects Through Some Containers	• Very Sensitive to Extreme Environmental Changes	• Level Sensing
Ultrasonic	• Senses all Materials	• Sensitive to Temperature Changes.	• Level Control • Doors • Anti-Collision

١- ٣ - Electromagnetic Relay : -

relay is an electrically operated switch. Current flowing through the coil of the relay creates a magnetic field which attracts a lever and changes the switch contacts. The coil current can be on or off so relays have two switch positions and they are double throw (changeover) switches.

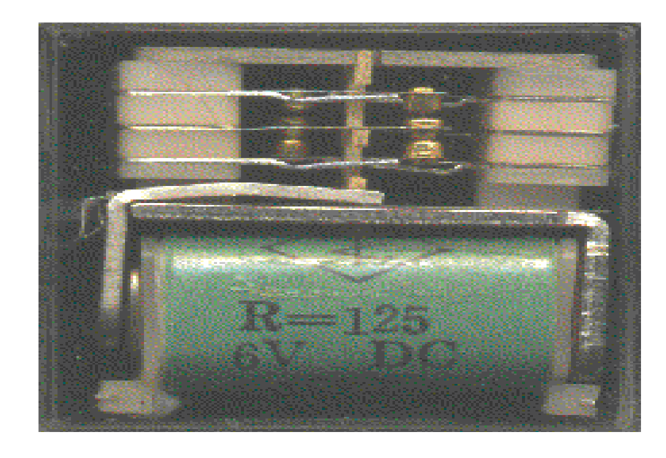


Circuit symbol for a relay

Relays allow one circuit to switch a second circuit which can be completely separate from the first. For example a low voltage battery circuit can use a relay to switch a ٢٣٠V AC mains circuit. There is no electrical connection inside the relay between the two circuits, the link is magnetic and mechanical. The coil of a relay passes a relatively large current, typically ٣٠mA for a ١٢V relay, but it can be as much as ١٠٠mA for relays designed to operate from lower voltages

Relays are usually SPDT or DPDT but they can have many more sets of switch contacts, for example relays with ٤ sets of changeover contacts are readily available.

The animated picture shows a working relay with its coil and switch contacts. You can see a lever on the left being attracted by magnetism when the coil is switched on. This lever moves the switch contacts. There is one set of contacts (SPDT) in the foreground and another behind them, making the relay DPDT.





Relays

The relay's switch connections are usually labeled COM, NC and NO:

- **COM = Common, always connect to this, it is the moving part of the switch.**
- **NC = Normally Closed, COM is connected to this when the relay coil is off.**
- **NO = Normally Open, COM is connected to this when the relay coil is on.**

- **Connect to COM and NO if you want the switched circuit to be on when the relay coil is on.**
- **Connect to COM and NC if you want the switched circuit to be on when the relay coil is off.**

Choosing a relay : -

You need to consider several features when choosing a relay:

١. **Physical size and pin arrangement**
If you are choosing a relay for an existing PCB you will need to ensure that its dimensions and pin arrangement are suitable. You should find this information in the supplier's catalogue.

٢. **Coil voltage**
The relay's coil voltage rating and resistance must suit the circuit powering the relay coil. Many relays have a coil rated for a ١٢V supply but ٥V and ٢٤V relays are also readily available. Some relays operate perfectly well with a supply voltage which is a little lower than their rated value.

٣. **Coil resistance**
The circuit must be able to supply the current required by the relay coil. You can use Ohm's law to calculate the current:

$$\text{Relay coil current} = \frac{\text{supply voltage}}{\text{coil resistance}}$$

٤. **For example: A ١٢V supply relay with a coil resistance of ٤٠٠Ω passes a curre ٣٠mA.**

٥. **Switch ratings (voltage and current)**
   The relay's switch contacts must be suitable for the circuit they are to control. You will need to check the voltage and current ratings. Note that the voltage rating is usually higher for AC, for example: "٥A at ٢٤V DC or ١٢٥V AC".

٦. **Switch contact arrangement (SPDT, DPDT etc)**
   Most relays are SPDT or DPDT which are often described as "single pole changeover" (SPCO) or "double pole changeover" (DPCO).

example).

Advantages of relays:

- **Relays can switch AC and DC, transistors can only switch DC.**
- **Relays can switch high voltages, transistors cannot.**
- **Relays are a better choice for switching large currents (> ٥A).**
- **Relays can switch many contacts at once.**

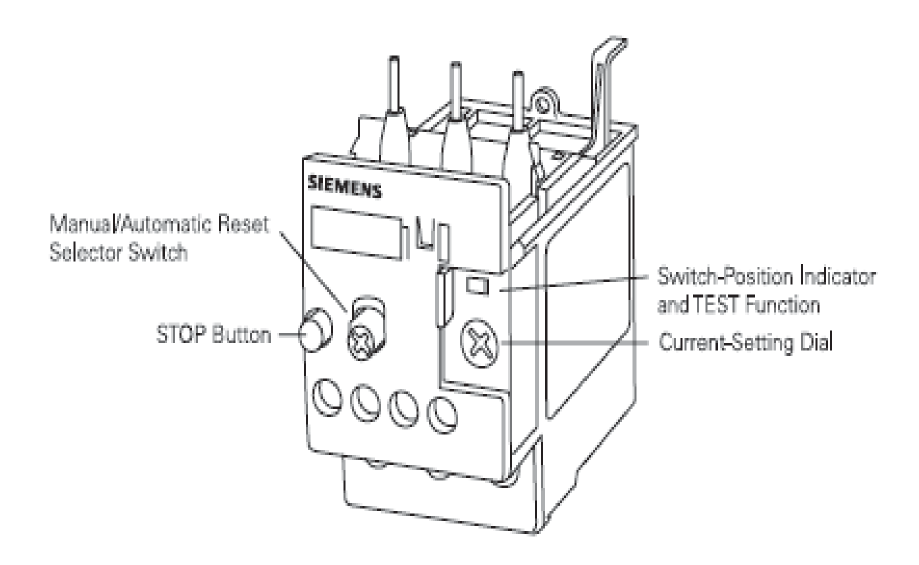
Disadvantages of relays:

- **Relays are bulkier than transistors for switching small currents.**
- **Relays cannot switch rapidly (except reed relays), transistors can switch many times per second.**
- **Relays use more power due to the current flowing through their coil.**
- **Relays require more current than many chips can provide, so a low power transistor may be needed to switch the current for the relay's coil.**

Relays can generate a very high voltage across the coil when switched off. This can damage other components in the circuit. To prevent this a diode is connected across the coil. The cathode of the diode is connected to the most positive end of the coil.

- **Overload Relay**

A device used to protect a motor from damage resulting from an *overcurrent*.



- **Overcurrent**

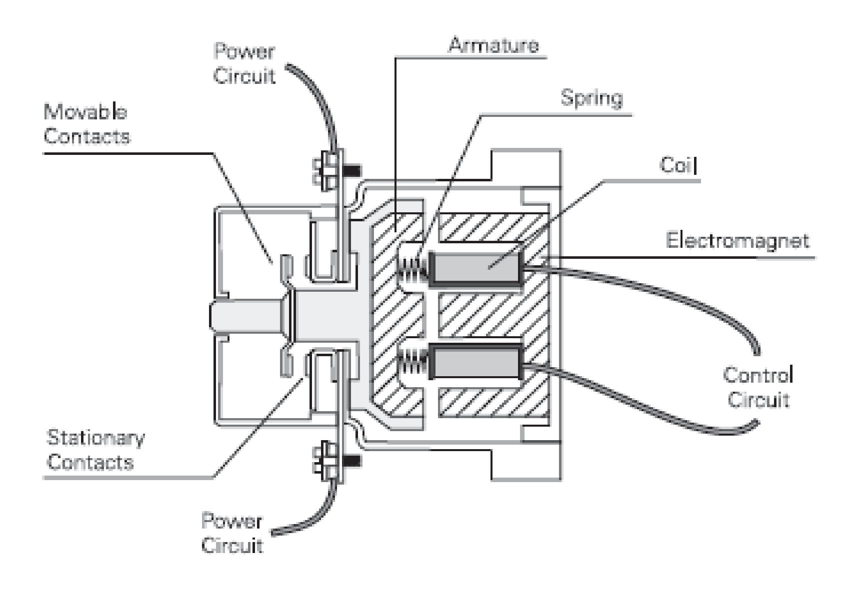
A *current* in excess of the rated current for a device or *conductor*. An overcurrent can result from an *overload, short circuit,* or *ground fault*.

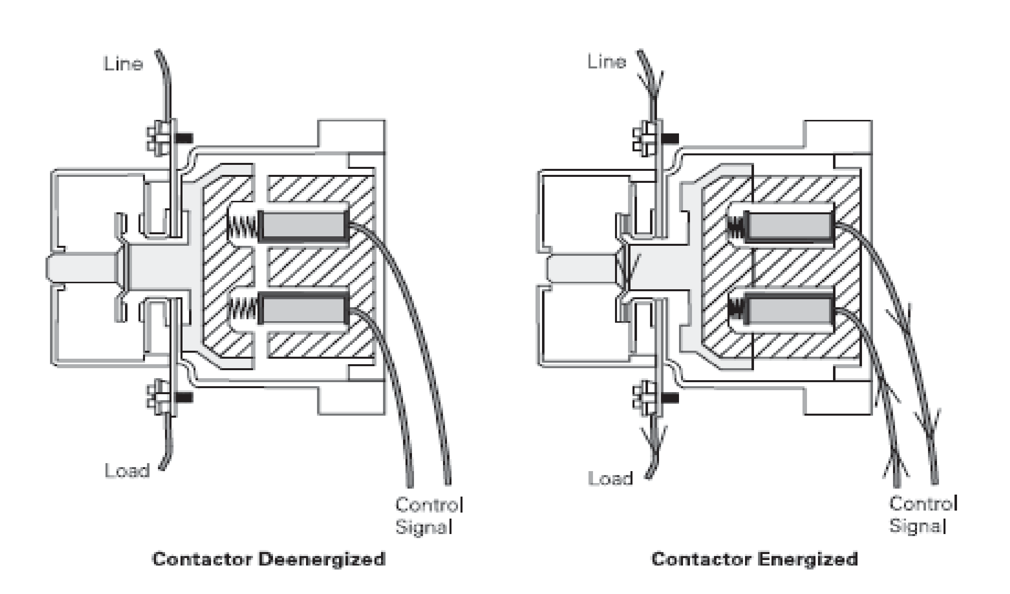
- **Overload**

Can refer to an operating condition in excess of a full-load rating or a *current* high enough to cause damage if it is present long enough. An overload does not refer to a *short circuit* or *ground fault*.

١.٤- Contactor :-

A device used to energize and de-energize an electrical circuit.



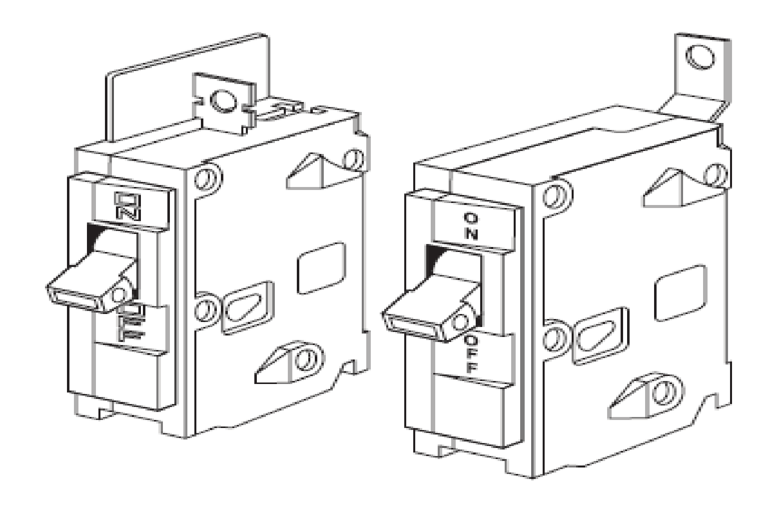


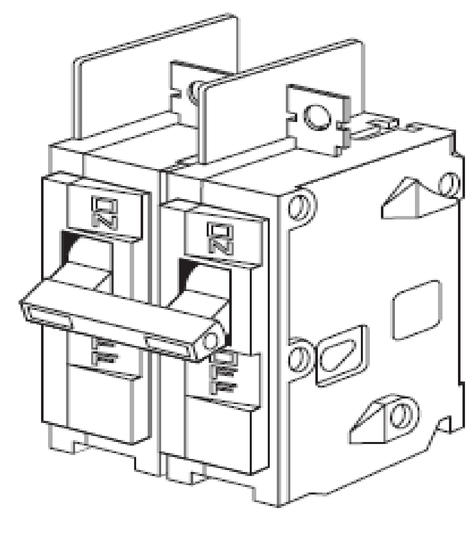
Contactor Deenergized Contactor Energized

٢- Circuit Breakers : -

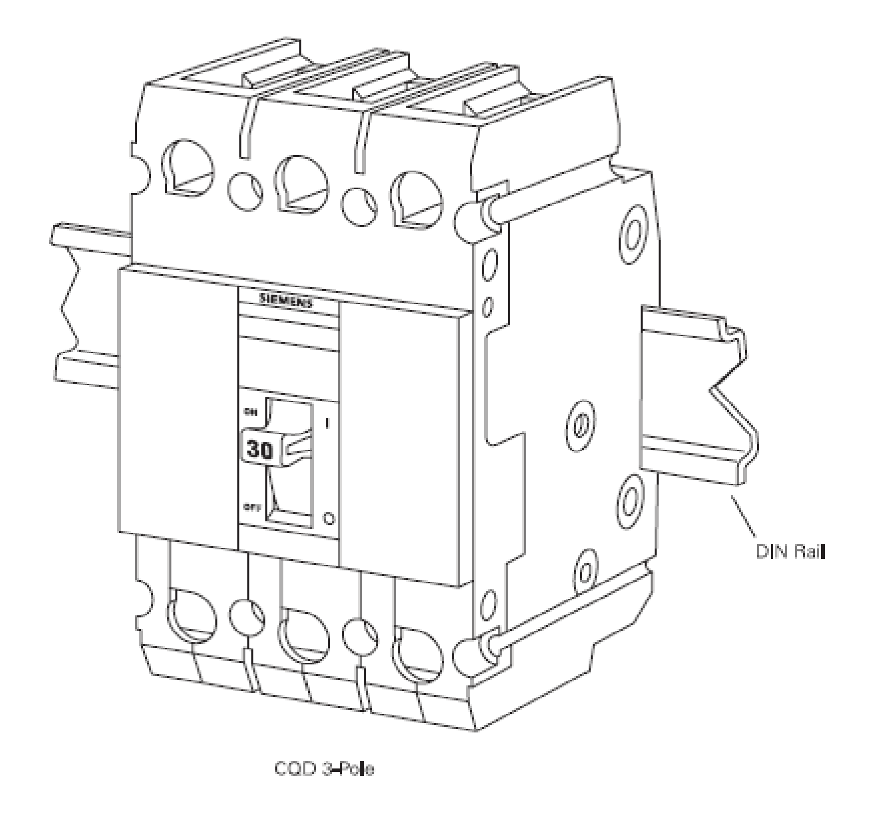
Residential and Commercial Circuit Breakers : -

There are several circuit breakers that are used in residential, commercial and light industrial applications. These circuit breakers are normally plug-in or bolt-on types that mount in load centers or panel boards . Other types are also available, for example, circuit breakers that mount on a DIN rail. There are several variations of circuit breakers, and this section will attempt to explain the most popular of them.





BQ 2-Pole

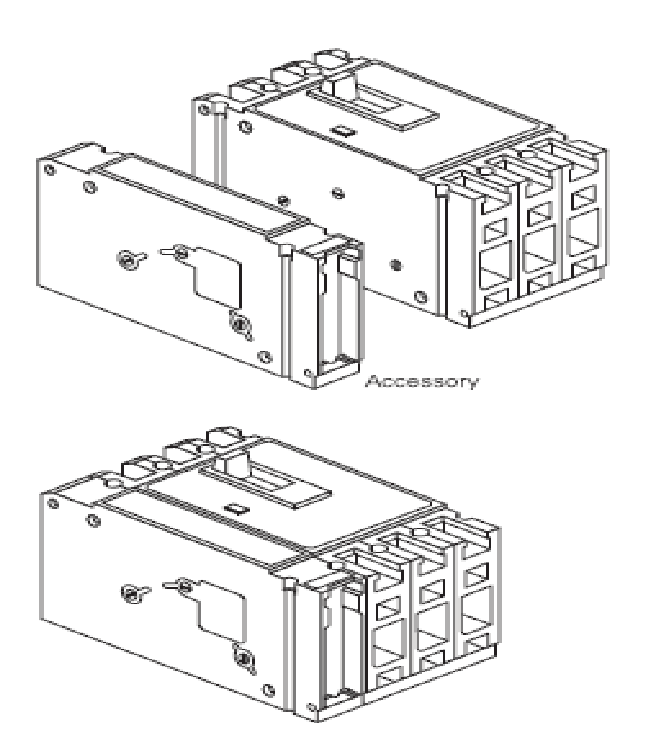


CQD 3-Pole

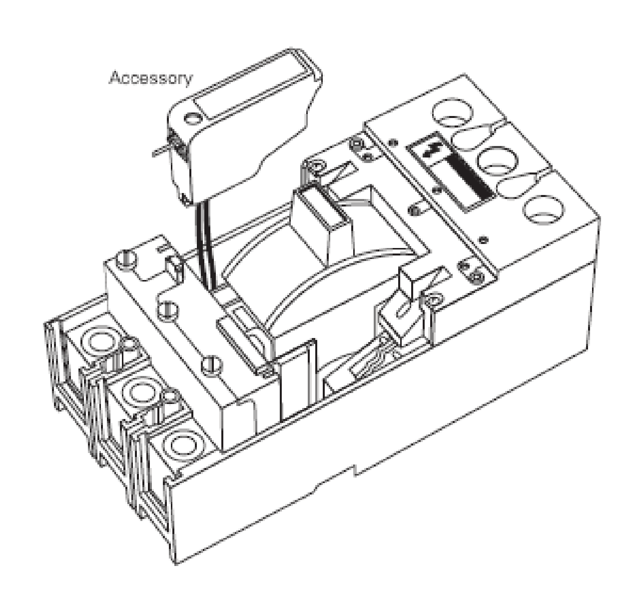
- **<u>Internal Accessories : -</u>**

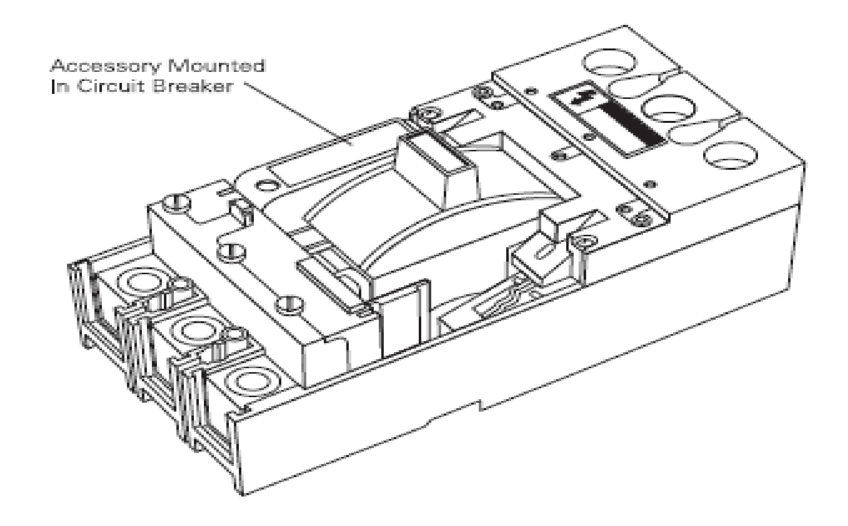
An accessory is an addition that adds to the performance of a circuit breaker or adapts the circuit breaker for specific application requirements. Various accessories are available for Siemens molded case circuit breakers. Internal accessories are used to modify a breaker's performance. The four internal accessories are shunt trip, under voltage trip, auxiliary switches, and bell alarm.

The circuit breaker internal accessories are mounted on the side of the circuit breaker as shown in the following illustration.



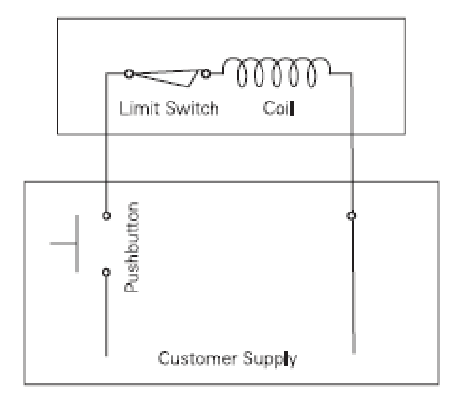
To mount internal accessories in circuit breakers, the cover is removed and the accessories installed as shown in the following illustrations.





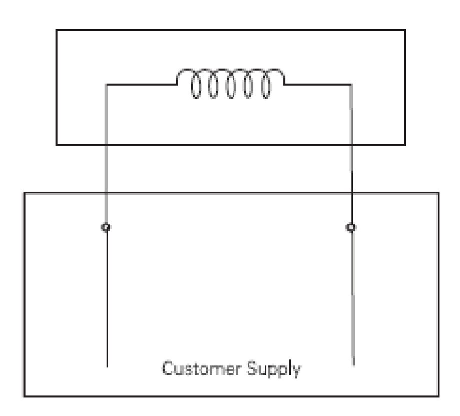
- **Shunt Trip : -**

It is sometimes necessary to trip a breaker from a remote location. For example, if someone were to get caught in a piece of machinery, anyone can push a "panic button" tripping the breaker. One or all critical circuit breakers may be tripped at the push of a button from a distant control point by use of a shunt trip device. The shunt trip device consists of a coil in series with a limit switch. When the circuit breaker contacts are closed the limit switch is closed. Depressing a customer supplied pushbutton energizes the shunt trip coil, causing the breaker's mechanical latch to disengage the trip mechanism and opening the circuit breaker's contacts. When the circuit breaker's contacts open the limit switch also opens, removing power from the shunt trip coil. As with any trip the breaker must be reset manually.



- **Under voltage Trip : -**

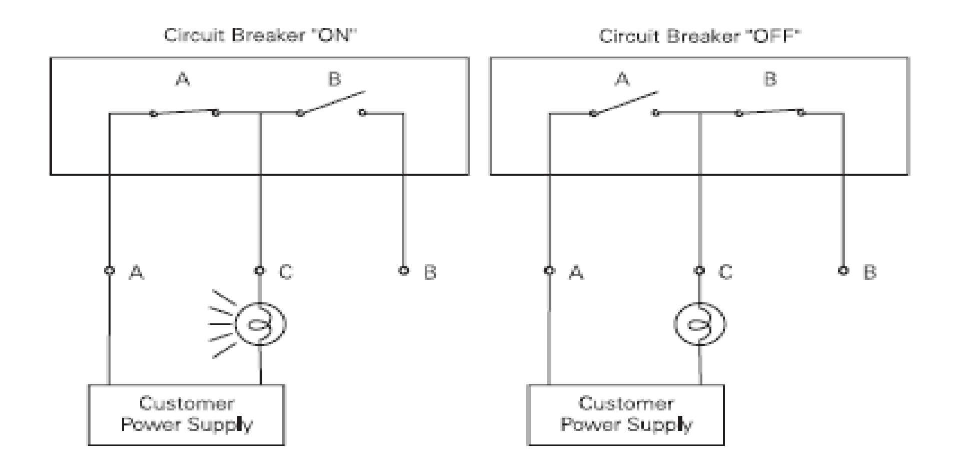
The under voltage trip device is designed to automatically trip the circuit breaker when the supply voltage drops to a low value (٣٥-٧٠٪ of nominal line voltage). The device also prevents the circuit breaker from being reclosed until the supply voltage returns to at least ٨٥٪ of its normal level.



- **Auxiliary Switch : -**

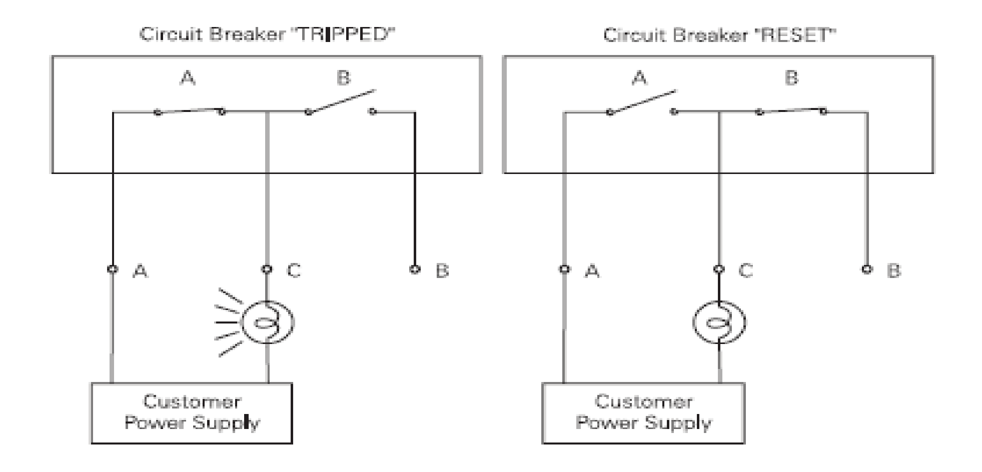
An auxiliary switch consists of one set of normally open and one set of normally closed contacts. Contact "A" is open when the circuit breaker is in the "Off" or "Tripped" conditions. Contact "B" is closed when the circuit breaker is in the "Off" or "Tripped" conditions. This accessory is used to indicate whether a circuit breaker is on or off from a remote location. For example, in the following illustration an indicator light is connected to a customer's power supply through contact "A".

When the circuit breaker is switched on, the light illuminates, indicating the circuit breaker's contacts are closed and the breaker is supplying power to the load. When the circuit breaker is switched off, contact "A" opens, turning the indicator light off.



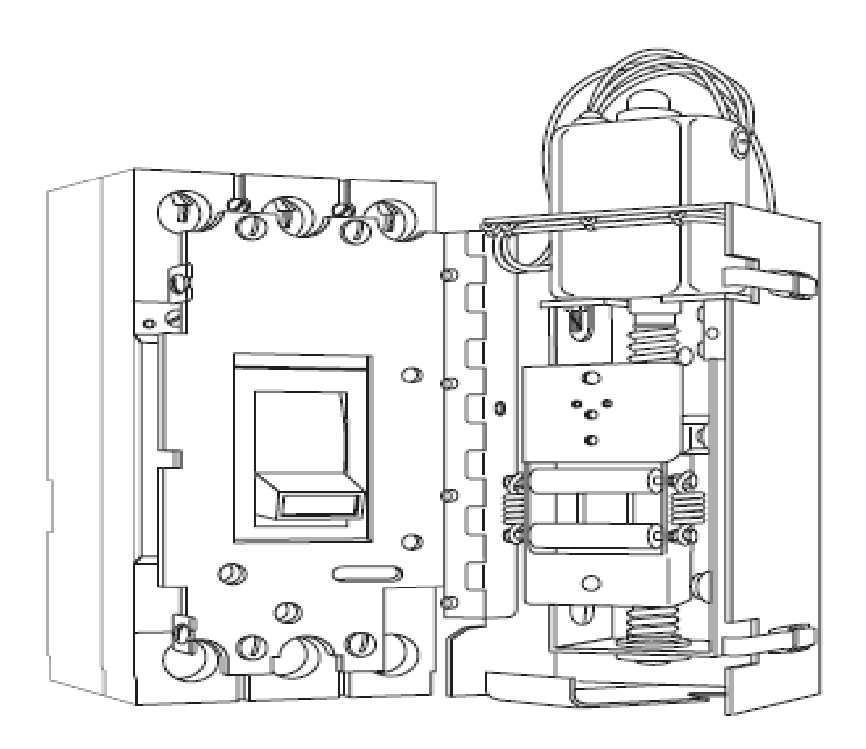
- **Bell Alarm Switch : -**

The bell alarm switch differs from the auxiliary switch in that it only functions when the circuit breaker trips. Opening and closing the circuit breaker by means of the operating handle does not affect the position of the alarm contacts. The "A" contact closes when the circuit breaker trips. The "B" contact opens when the circuit breaker trips. A horn or indicator light can be used to indicate the circuit breaker has tripped.



- **<u>Electric Motor Operator : -</u>**

It is designed to Motor Operator open, close and reset a circuit breaker by remote control. It is mounted on the face of the circuit breaker so that it can engage the breaker's operating handle. The built-in motor is connected to remote pushbuttons. Pressing the "ON" pushbutton causes the electric motor to move the circuit breaker to the "ON" position. Pressing the "OFF" pushbutton causes the electric motor to move the circuit breaker to the "OFF" position. To reset the circuit breaker from the tripped position, press the "OFF" pushbutton to move the handle to the "OFF" position. Then press the "ON" pushbutton to the close the breaker contacts.



Digital Control

١- Analogue systems : -

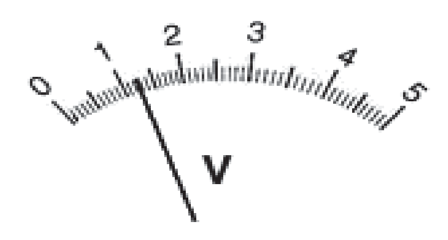
Analogue systems process analogue signals which can take any value within a range, for example the output from an LDR (light sensor) or a microphone.

An audio amplifier is an example of an analogue system. The amplifier produces an output voltage which can be any value within the range of its power supply.

Analogue signal

An analogue meter can display any value within the range available on its scale. However, the precision of readings is limited by our ability to read them. For example the meter on the right shows ١.٢٥V because the pointer is estimated to be half way between ١.٢ and ١.٣. The analogue meter can show any value between ١.٢ and ١.٣ but we are unable to read the scale more precisely than about half a division.

Analogue meter display



٢ - Digital systems : -

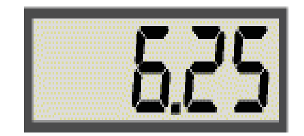
Digital systems process digital signals which can take only a limited number of values (discrete steps), usually just two values are used: the positive supply voltage (+Vs) and zero volts (٠V).

Digital systems contain devices such as logic gates, flip-flops, shift registers and counters. A computer is an example of a digital system.

Digital (logic) signal

A digital meter can display many values, but not every value within its range. For example the display on the right can show ٦.٢٥ and ٦.٢٦ but not a value between them. This is not a problem because digital meters normally have sufficient digits to show values more precisely than it is possible to read an analogue display.

Digital meter display



٣- Logic signals : -

Most digital systems use the simplest possible type of signal which has just two values. This type of signal is called a logic signal because the two values (or states) can be called true and false. Normally the positive supply voltage +Vs represents true

and ٠V represents false. Other labels for the true and false states are shown in the table down .

Noise is relatively easy to eliminate from digital signals because it is easy to distinguish from the desired signal which can only have particular values. For example: if the signal is meant to be +٥V (true) or ٠V (false), noise of up to ٢.٥V can be eliminated by treating all voltages greater than ٢.٥V as true and all voltages less than ٢.٥V as false.

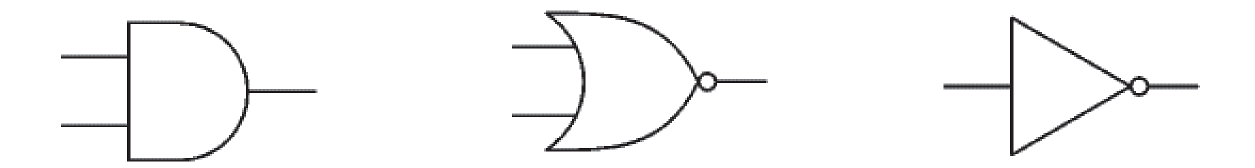
Logic states	
True	**False**
١	٠
High	**Low**
+Vs	**٠V**
On	**Off**

Gates are identified by their function: NOT, AND, NAND, OR, NOR, EX-OR and EX-NOR. Capital letters are normally used to make it clear that the term refers to a logic gate. the logic gates are not always required because simple logic functions can be performed with switches or diodes .

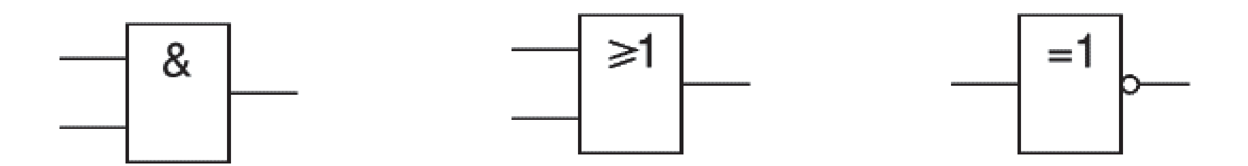
٣-١ Logic gate symbols : -

There are two series of symbols for logic gates:

١- **The traditional symbols have distinctive shapes making them easy to recognize so they are widely used in industry and education.**

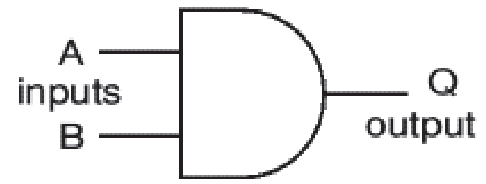


٢- **The IEC (International Electro technical Commission) symbols are rectangles with a symbol inside to show the gate function. They are rarely used despite their official status, but you may need to know them .**



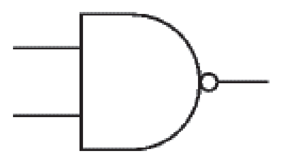
٣-٢ Inputs and outputs : -

Gates have two or more inputs, except a NOT gate which has only one input. All gates have only one output. Usually the letters A, B, C and so on are used to label inputs, and Q is used to label the output. On this page the inputs are shown on the left and the output on the right.



٣-٣ The inverting circle (o) : -

Some gate symbols have a circle on their output which means that their function includes inverting of the output. It is equivalent to feeding the output through a NOT gate. For example the NAND (Not AND) gate symbol shown on the right is the same as an AND gate symbol but with the addition of an inverting circle on the output.



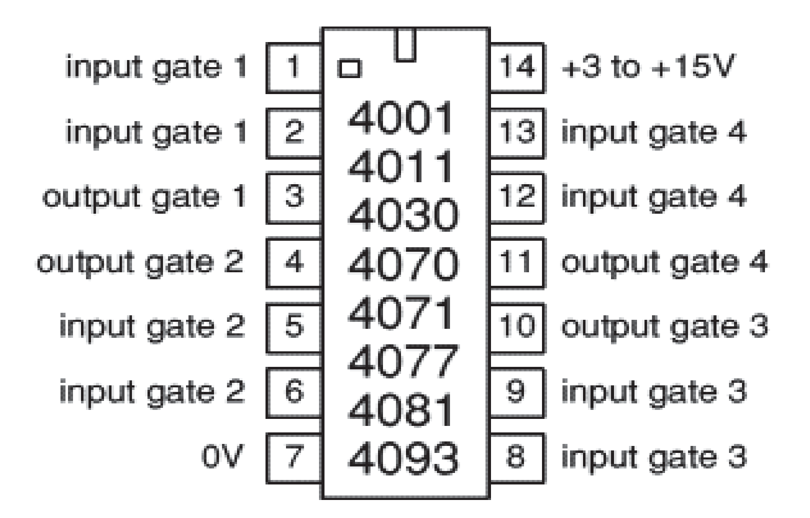
٣-٤ Truth tables : -

A truth table is a good way to show the function of a logic gate. It shows the output states for every possible combination of input states. The symbols ٠ (false) and ١ (true) are usually used in truth tables. The example truth table on the right shows the inputs and output of an AND gate.

Input A	Input B	Output Q
٠	٠	٠
٠	١	٠
١	٠	٠
١	١	١

٣-٥ Logic ICs : -

Logic gates are available on special ICs (chips) which usually contain several gates of the same type, for example the ٤٠٠١ IC contains four ٢-input NOR gates. There are several families of logic ICs and they can be split into two groups:

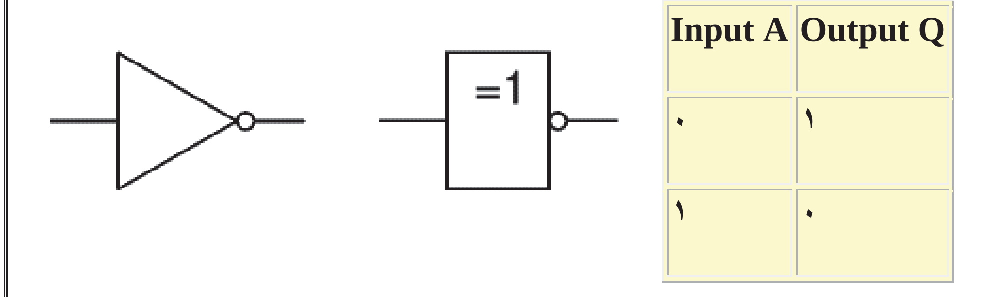


٣-٦ Type of logic gates : -

٣-٦ -١ NOT gate (inverter) : -

The output Q is true when the input A is NOT true, the output is the inverse of the input: Q = NOT A

A NOT gate can only have one input. A NOT gate is also called an inverter.



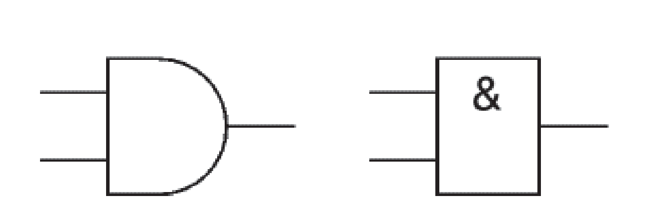
Input A	Output Q
٠	١
١	٠

Traditional symbol IEC symbol Truth Table

٣-٦-٢ AND gate : -

The output Q is true if input A AND input B are both true: Q = A AND B

An AND gate can have two or more inputs, its output is true if all inputs are true.

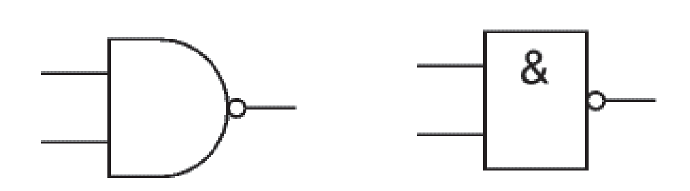


Input A	Input B	Output Q
٠	٠	٠
٠	١	٠
١	٠	٠
١	١	١

Traditional symbol IEC symbol Truth Table

٣-٦-٣ NAND gate (NAND = Not AND) : -

This is an AND gate with the output inverted, as shown by the 'o' on the output. The output is true if input A AND input B are NOT both true: Q = NOT (A AND B) A NAND gate can have two or more inputs, its output is true if NOT all inputs are true.

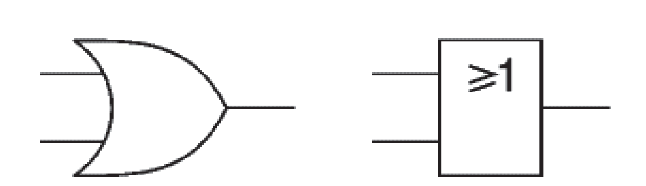


Input A	Input B	Output Q
٠	٠	١
٠	١	١
١	٠	١
١	١	٠

Traditional symbol IEC symbol Truth Table

٣-٦-٤ OR gate : -

The output Q is true if input A OR input B is true (or both of them are true): Q = A OR B . An OR gate can have two or more inputs, its output is true if at least one input is true.

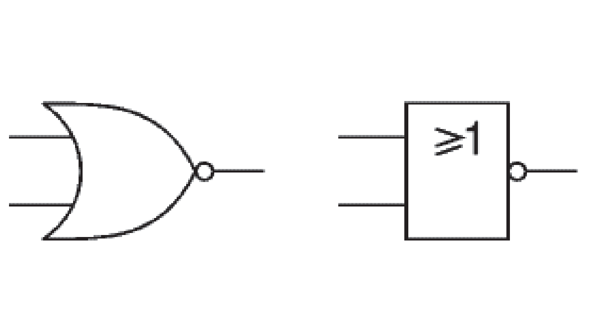


Input A	Input B	Output Q
٠	٠	٠
٠	١	١
١	٠	١
١	١	١

Traditional symbol IEC symbol Truth Table

٣-٦-٥ NOR gate (NOR = Not OR)

This is an OR gate with the output inverted, as shown by the 'o' on the output. The output Q is true if NOT inputs A OR B are true: Q = NOT (A OR B) A NOR gate can have two or more inputs, its output is true if no inputs are true.



Input A	Input B	Output Q
٠	٠	١
٠	١	٠
١	٠	٠
١	١	٠

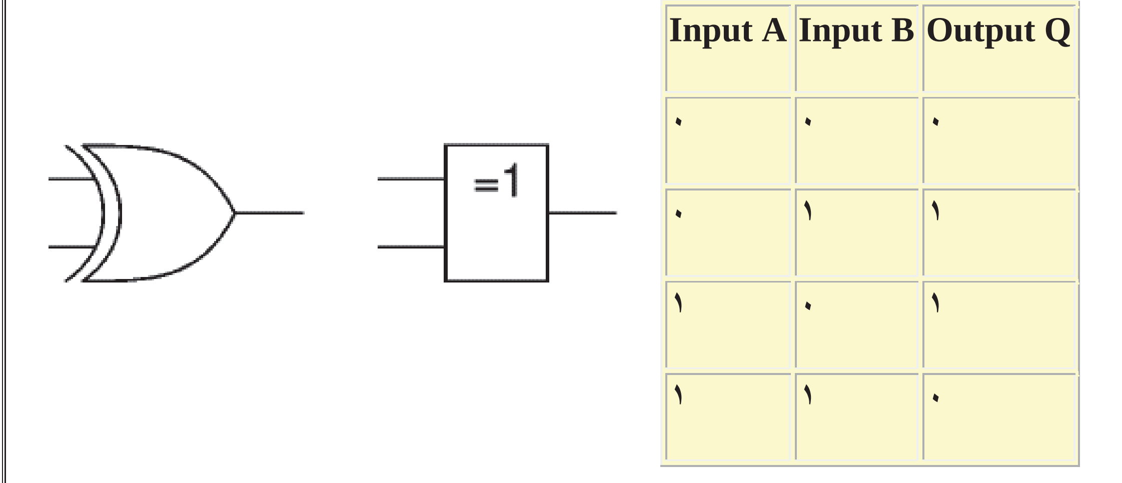
Traditional symbol IEC symbol Truth Table

٣-٦-٦ EX-OR (EXclusive-OR) gate : -

The output Q is true if either input A is true OR input B is true, but not when both of them are true: Q = (A AND NOT B) OR (B AND NOT A) This is like an OR gate but excluding both inputs being true.

The output is true if inputs A and B are DIFFERENT.

EX-OR gates can only have ٢ inputs.



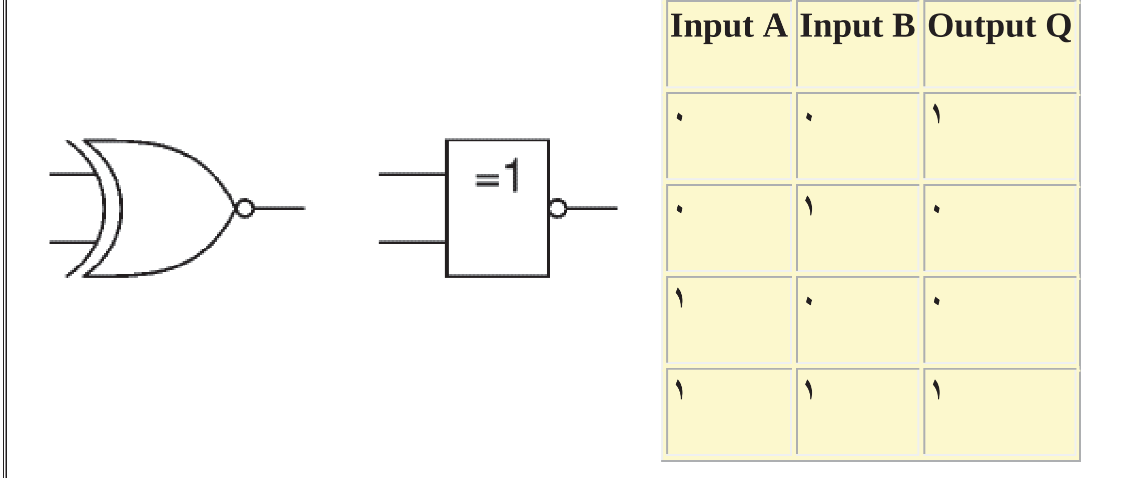
Input A	Input B	Output Q
٠	٠	٠
٠	١	١
١	٠	١
١	١	٠

Traditional symbol IEC symbol Truth Table

٣-٦-٧ EX-NOR (EXclusive-NOR) gate : -

This is an EX-OR gate with the output inverted, as shown by the 'o' on the output. The output Q is true if inputs A and B are the SAME (both true or both false): Q = (A AND B) OR (NOT A AND NOT B)

EX-NOR gates can only have ٢ inputs.



Input A	Input B	Output Q
٠	٠	١
٠	١	٠
١	٠	٠
١	١	١

Traditional symbol IEC symbol Truth Table

٣-٧ Summary truth tables : -

The summary truth tables below show the output states for all types of ٢-input and ٣-input gates.

Summary for all ٢-input gates							
Inputs		**Output of each gate**					
A	**B**	**AND**	**NAND**	**OR**	**NOR**	**EX-OR**	**EX-NOR**
٠	٠	٠	١	٠	١	٠	١
٠	١	٠	١	١	٠	١	٠
١	٠	٠	١	١	٠	١	٠
١	١	١	٠	١	٠	٠	١

Summary for all ٣-input gates						
Inputs			**Output of each gate**			
A	**B**	**C**	**AND**	**NAND**	**OR**	**NOR**
٠	٠	٠	٠	١	٠	١
٠	٠	١	٠	١	١	٠
٠	١	٠	٠	١	١	٠
٠	١	١	٠	١	١	٠
١	٠	٠	٠	١	١	٠
١	٠	١	٠	١	١	٠
١	١	٠	٠	١	١	٠
١	١	١	١	٠	١	٠

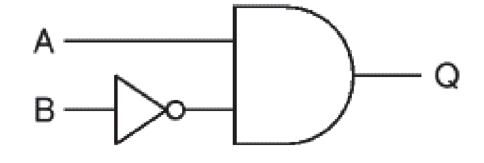
Note that EX-OR and EX-NOR gates can only have ٢ inputs.

٣-٨ Combinations of logic gates : -

Logic gates can be combined to produce more complex functions. They can also be combined to substitute one type of gate for another.

For example to produce an output Q which is true only when input A is true and input B is false, as shown in the truth table on the right, we can combine a NOT gate and an AND gate like this:

Q = A AND NOT B

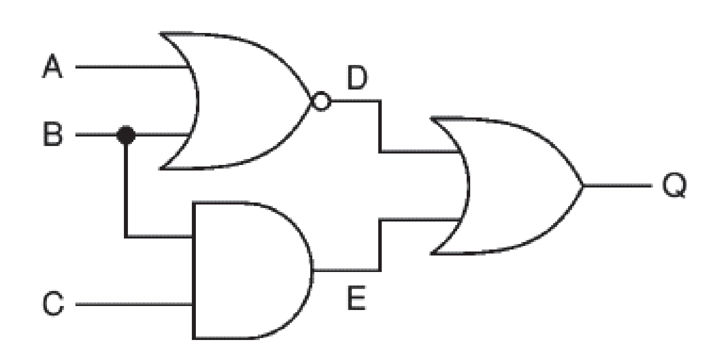


Input A	Input B	Output Q
٠	٠	٠
٠	١	٠
١	٠	١
١	١	٠

Working out the function of a combination of gates : -

Truth tables can be used to work out the function of a combination of gates.

For example the truth table on the right show the intermediate outputs D and E as well as the final output Q for the system shown below.



Inputs			Outputs		
A	B	C	D	E	Q
٠	٠	٠	١	٠	١
٠	٠	١	١	٠	١
٠	١	٠	٠	٠	٠
٠	١	١	٠	١	١
١	٠	٠	٠	٠	٠
١	٠	١	٠	٠	٠
١	١	٠	٠	٠	٠
١	١	١	٠	١	١

D = NOT (A OR B)

E = B AND C

Q = D OR E = (NOT (A OR B)) OR (B AND C) .

٣-٩ Substituting one type of gate for another : -

Logic gates are available on ICs which usually contain several gates of the same type, for example four ٢-input NAND gates or three ٣-input NAND gates. This can be wasteful if only a few gates are required unless they are all the same type. To avoid using too many ICs you can reduce the number of gate inputs or substitute one type of gate for another.

٣-١٠ Reducing the number of inputs of logic gates : -

The number of inputs to a gate can be reduced by connecting two (or more) inputs together. The diagram shows a ٣-input AND gate operating as a ٢-input AND gate. Making a NOT gate from a NAND or NOR gate Reducing a NAND or NOR gate to just one input creates a NOT gate. The diagram shows this for a ٢-input NAND gate.



Any gate can be built from NAND or NOR gates As well as making a NOT gate, NAND or NOR gates can be combined to create any type of gate! This enables a circuit to be built from just one type of gate, either NAND or NOR. For example an AND gate is a NAND gate then a NOT gate (to undo the inverting function). Note that AND and OR gates cannot be used to create other gates because they lack the inverting (NOT) function.

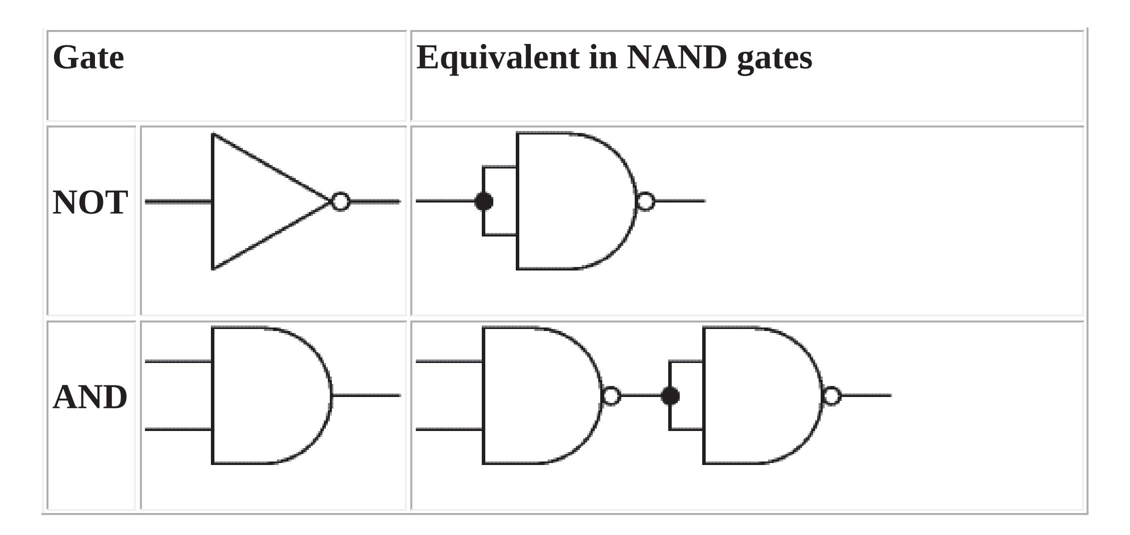
To change the type of gate, such as changing OR to AND, you must do three things:

- **Invert (NOT) each input.**
- **Change the gate type (OR to AND, or AND to OR)**
- **Invert (NOT) the output.**

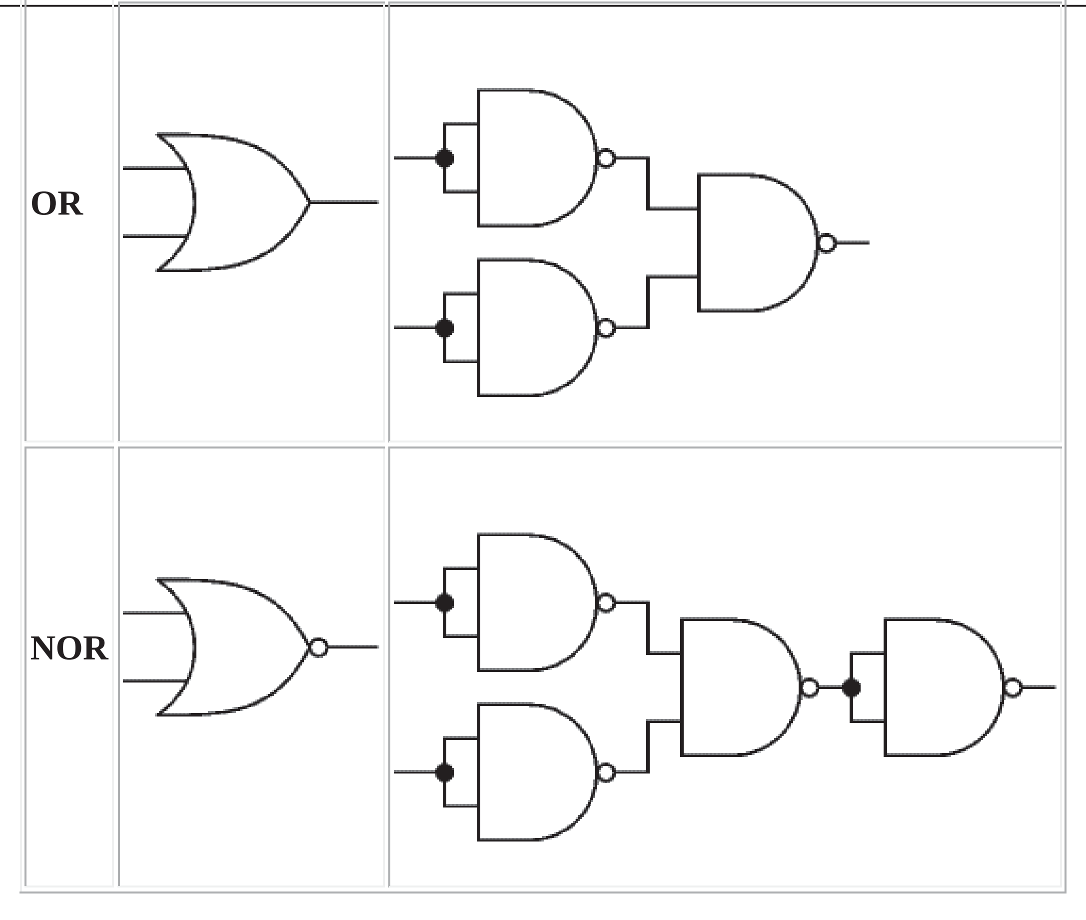
For example an OR gate can be built from NOTed inputs fed into a NAND (AND + NOT) gate.

<u>٣-١١ NAND gate equivalents : -</u>

The table below shows the NAND gate equivalents of NOT, AND, OR and NOR gates:

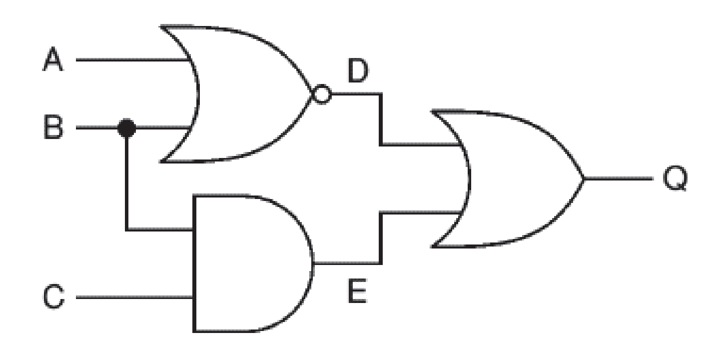


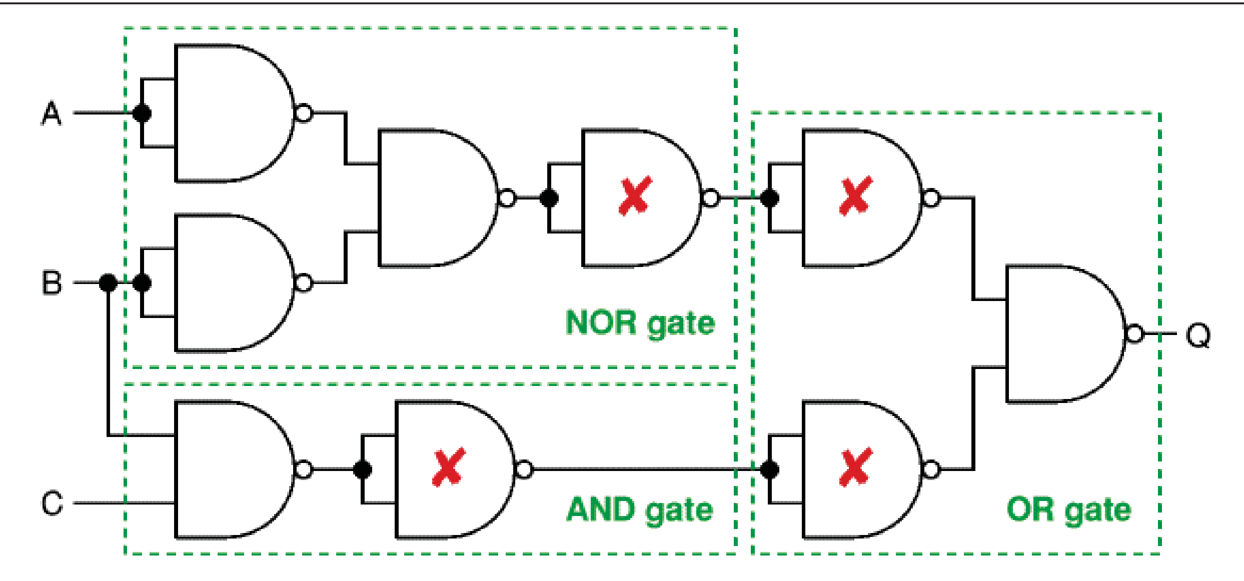
Gate		Equivalent in NAND gates
NOT		
AND		



<u>Substituting gates in an example logic system : -</u>

The original system has ٣ different gates: NOR, AND and OR. This requires three ICs (one for each type of gate).
To re-design this system using NAND gates only begin by replacing each gate with its NAND gate equivalent, as shown in the diagram below.

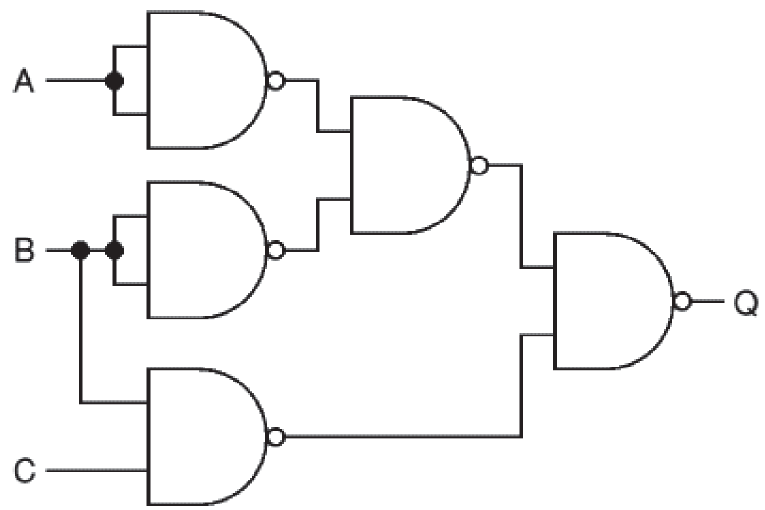




Then simplify the system by deleting adjacent pairs of NOT gates (marked X above). This can be done because the second NOT gate cancels the action of the first.

The final system is shown on the right. It has five NAND gates and requires two ICs (with four gates on each IC). This is better than the original system which required three ICs (one for each type of gate).

Substituting NAND (or NOR) gates does not always increase the number of gates, but when it does (as in this example) the increase is usually only one or two gates. The real benefit is reducing the number of ICs required by using just one type of gate.



As we said before the logic gates are not always required because simple logic functions can be performed in conventional control system using switches or contact of relay or by electronic element such as diodes or transistor .

<u>٤- Integrated Circuit : -</u>

IC's, often called "chips", come in several shapes and sizes.

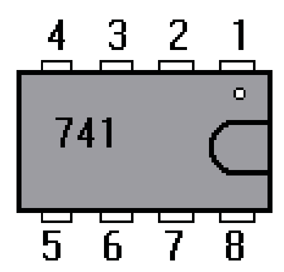
Most common are ٨, ١٤, or ١٦ pin dual in line (dil) chips.

IC's can be soldered directly into printed circuit boards, or may plug into sockets which have already been soldered into the board.

When soldering, ensure that the IC (or the socket) is the correct way round and that no pins have been bent underneath the body.

When fitting new IC's it is often necessary to bend the pins in slightly, in order to fit it into the board (or socket).

Some IC's are damaged by the static electricity which most people carry on their bodies. They should be stored in conductive foam or wrapped in tin foil. When handling them, discharge yourself periodically by touching some metalwork which is earthed, such as a radiator.



view from component side of pcb

٤-١ Linear integrated circuits : -

- **Handle analog signals.**
- **Important component " op-amp".**
- **Involve complex math (differentiation , integration,…).**
- **Bases of loop controls.**
- **Limited fine tuning of feedback systems**

٤-٢ Digital integrated circuits : -

- **Deal exclusively with binary signals.**
- **Process information through logic gates.**
- **Various logic families (CMOS, TTL, …).**
- **Logic symbols & Boolean algebra…design & analysis.**
- **Large ICs with enormous number of gates.**
- **Heard of microprocessors.**

٥- Number Systems :

Since a PLC is a computer, it stores information in the form of On or Off conditions (١ or ٠), referred to as binary digits (bits). Sometimes binary digits are used individually and sometimes they are used to represent numerical values.

٥-١ Decimal System

Various number systems are used by PLCs. All number systems have the same three characteristics: digits, base, weight. The decimal system, which is commonly used in

everyday life, has the following characteristics:

Ten digits ٠, ١, ٢, ٣, ٤, ٥, ٦, ٧, ٨, ٩

Base ١٠

Weights ١, ١٠, ١٠٠, ١٠٠٠, ...

٥-٢ Binary System

The binary system is used by programmable controllers. The binary system has the following characteristics:

Two digits ٠, ١

Base ٢

Weights Powers of base ٢ (١, ٢, ٤, ٨, ١٦, ...)

Most Significant Bit ↓ … Least Significant Bit ↓

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
0	0	0	1	1	0	0	0

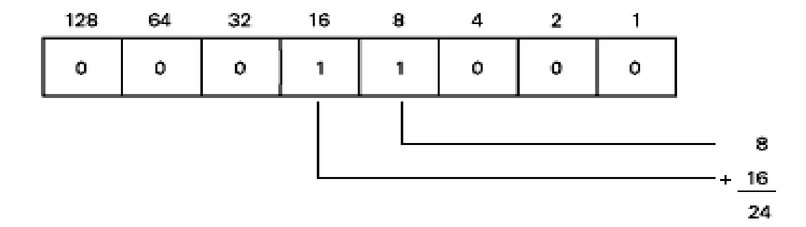
٥-٣ Converting Binary to Decimal

The following steps can be used to interpret a decimal number from a binary value.

١) Search from least to most significant bit for ١s.

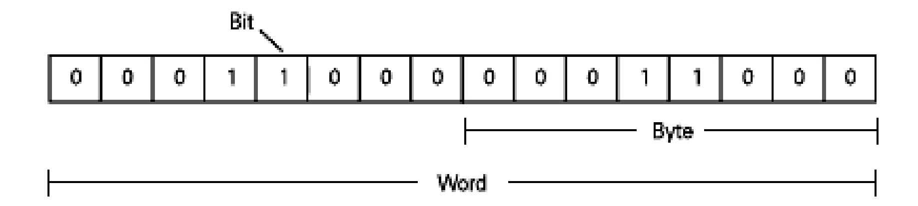
٢) Write down the decimal representation of each column containing a ١.

٣) Add the column values.



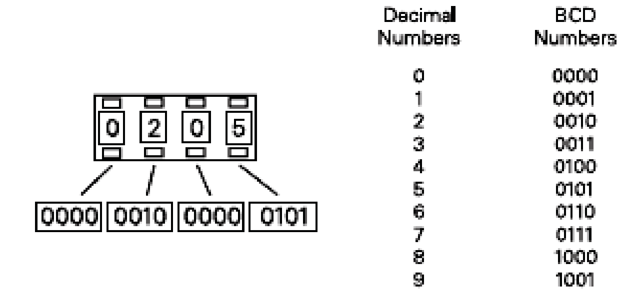
٥-٤ Bits, Bytes, and Words :

Each binary piece of data is a bit. Eight bits make up one byte. Two bytes, or ١٦ bits, make up one word.



٥-٥ BCD (Binary-Coded Decimal) :

Binary-Coded Decimal (BCD) are decimal numbers where each digit is represented by a four-bit binary number. BCD is commonly used with input and output devices. A thumbwheel switch is one example of an input device that uses BCD. The binary numbers are broken into groups of four bits, each group representing a decimal equivalent. A four-digit thumbwheel switch, like the one shown here, would control ١٦ (٤ x ٤) PLC inputs.



٥- ٦ Hexadecimal

Hexadecimal is another system used in PLCs. The hexadecimal system has the following characteristics:

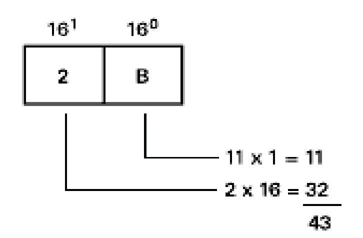
١٦ digits ٠, ١, ٢, ٣, ٤, ٥, ٦, ٧, ٨, ٩, A, B, C, D, E, F

Base ١٦

Weights Powers of base ١٦ (١, ١٦, ٢٥٦, ٤٠٩٦ ...)

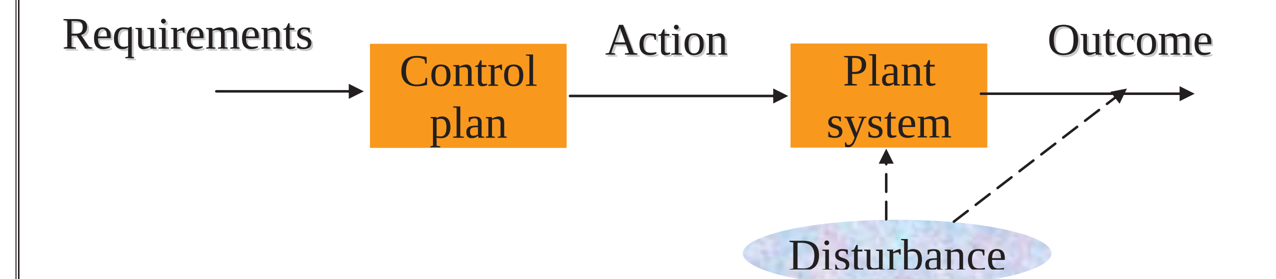
The ten digits of the decimal system are used for the first ten digits of the hexadecimal system. The first six letters of the alphabet are used for the remaining six digits.

A = ١٠	**D = ١٣**
B = ١١	**E = ١٤**
C = ١٢	**F = ١٥**



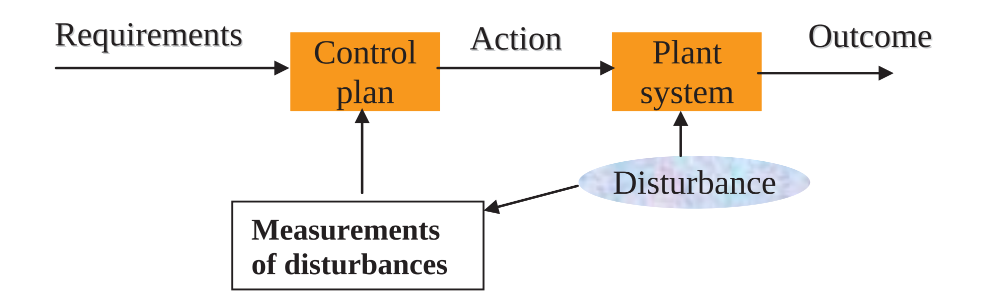
٦ - Control Strategies : -

٦- ١ Open Loop : -

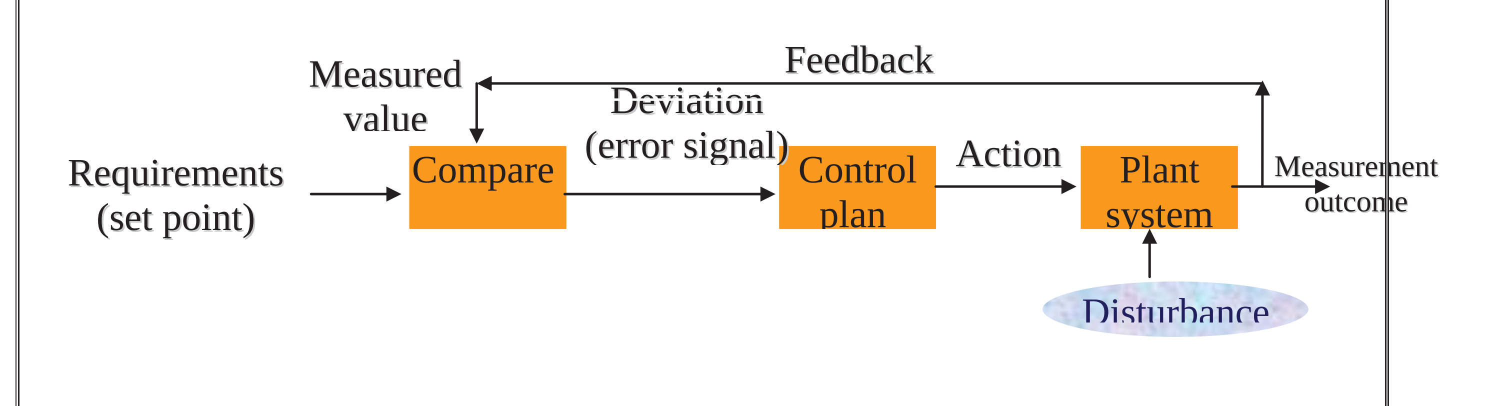


٦- ٢ Feed forward : -

Modification to plan taking account of disturbance



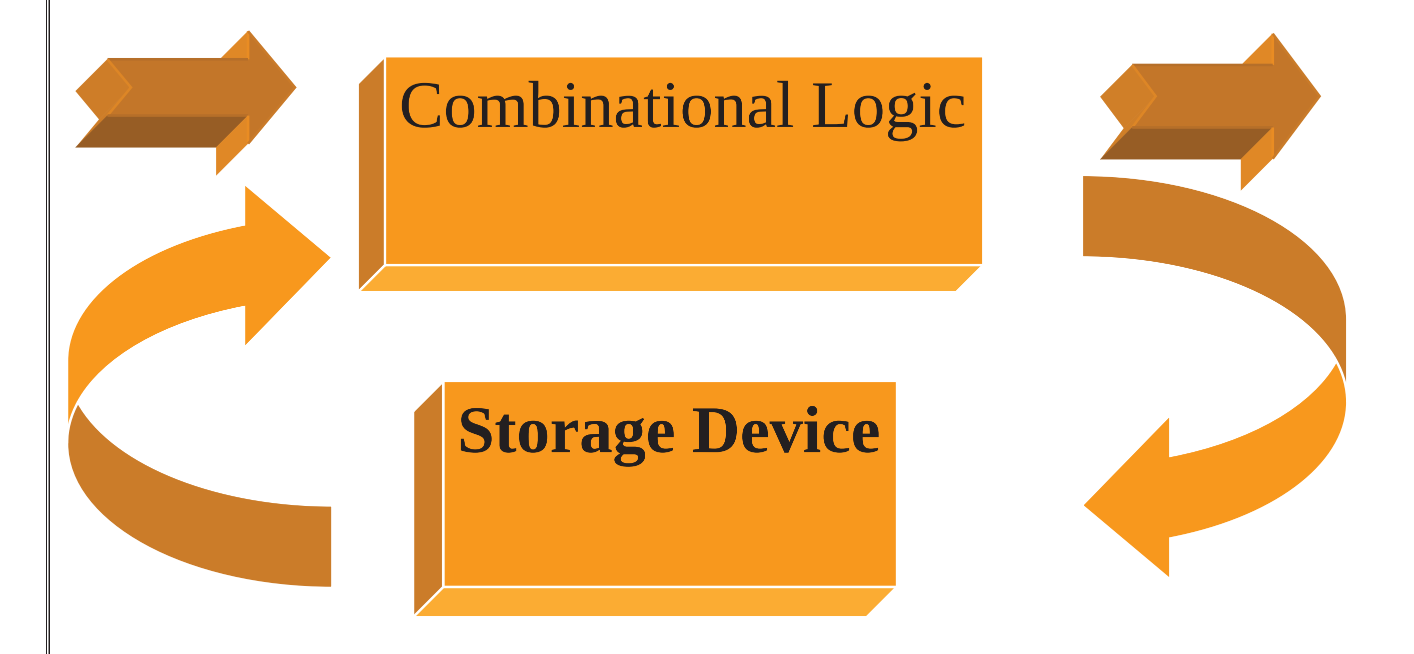
٦-٣ Closed Loop : -



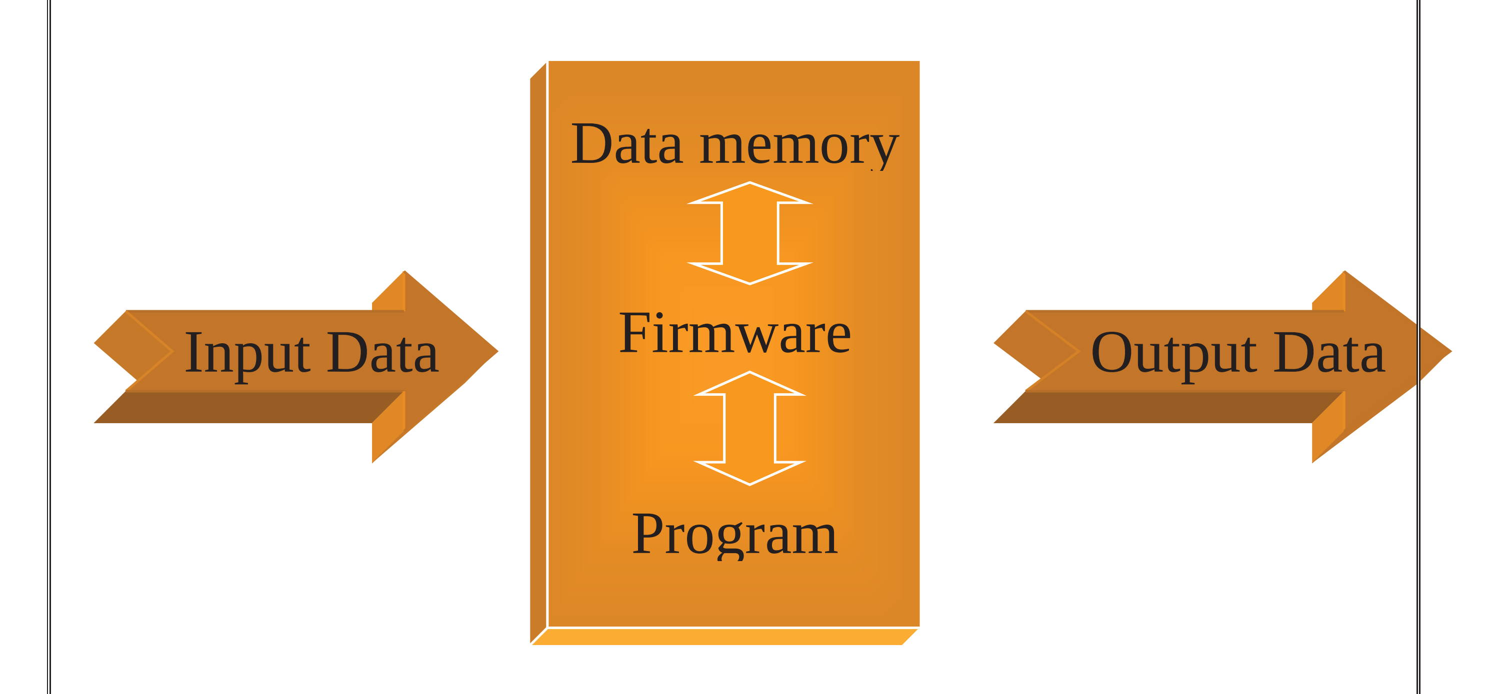
٦-٤ Sequential Control : -

basis of computer operation.

digital systems that have outputs dependent on previous system state .



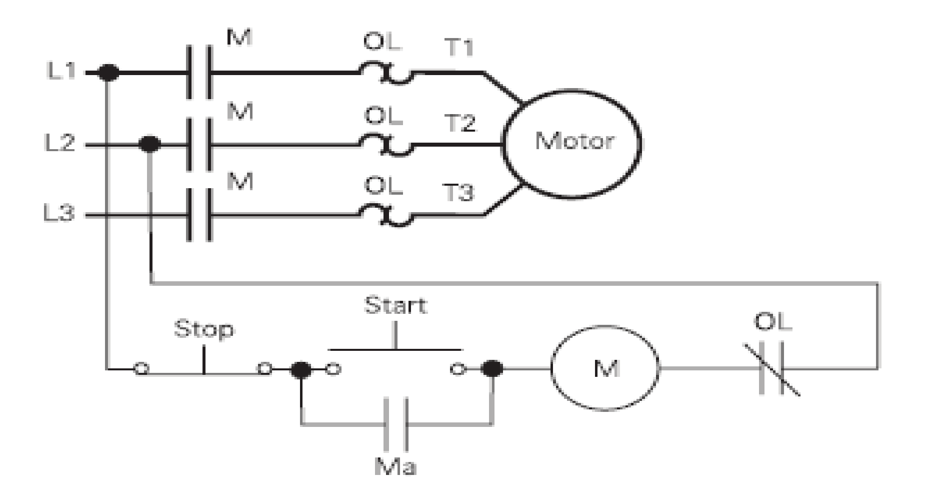
٦-٥ Programmable Computing Control Systems : -



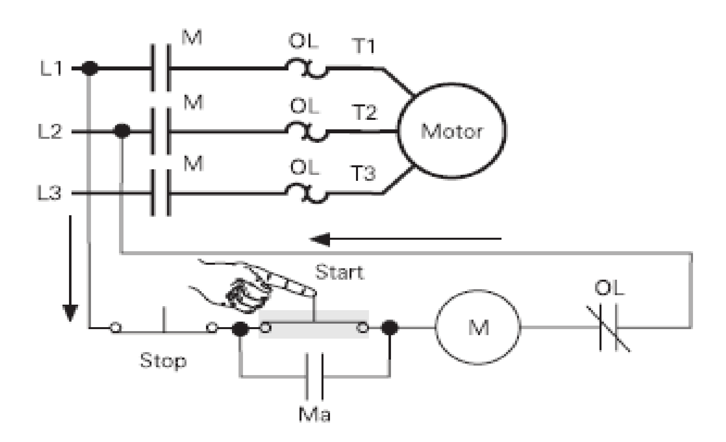
Applications (some of the application that will done through the course) :-

١ Motor Starter (Example ١) : -

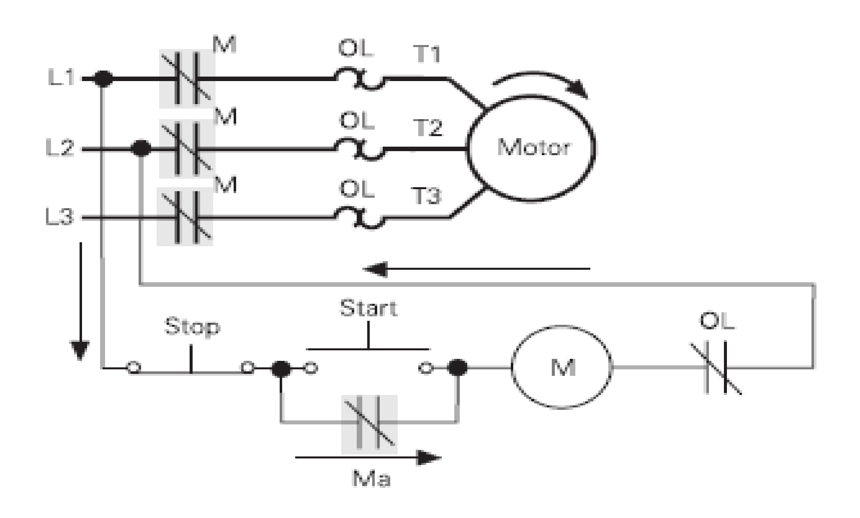
The following example involves a motor start and stop circuit. The line diagram illustrates how a normally open and a normally closed pushbutton might be used in a control circuit. In this example a motor started (M) is wired in series with a normally open momentary pushbutton (Start), a normally closed momentary pushbutton (Stop), and the normally closed contacts of an overload relay (OL).



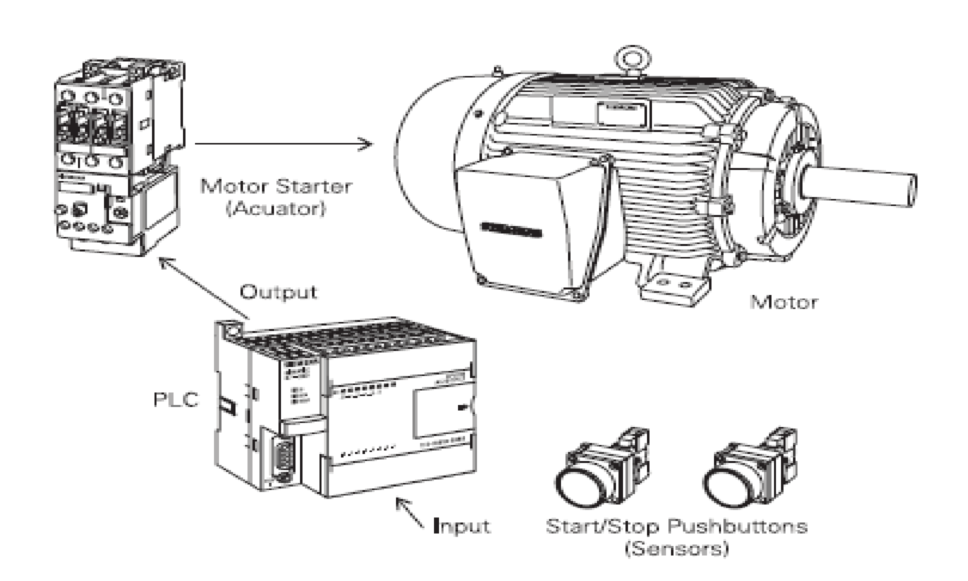
Momentarily depressing the Start pushbutton completes the path of current flow and energizes the motor starter (M).



This closes the associated M and Ma (auxiliary contact located in the motor starter) contacts. When the Start button is released a holding circuit exists to the M contactor through the auxiliary contacts Ma. The motor will run until the normally closed Stop button is depressed , or the overload relay opens the OL contacts , breaking the path of current flow to the motor starter and opening the associated M and Ma contacts.

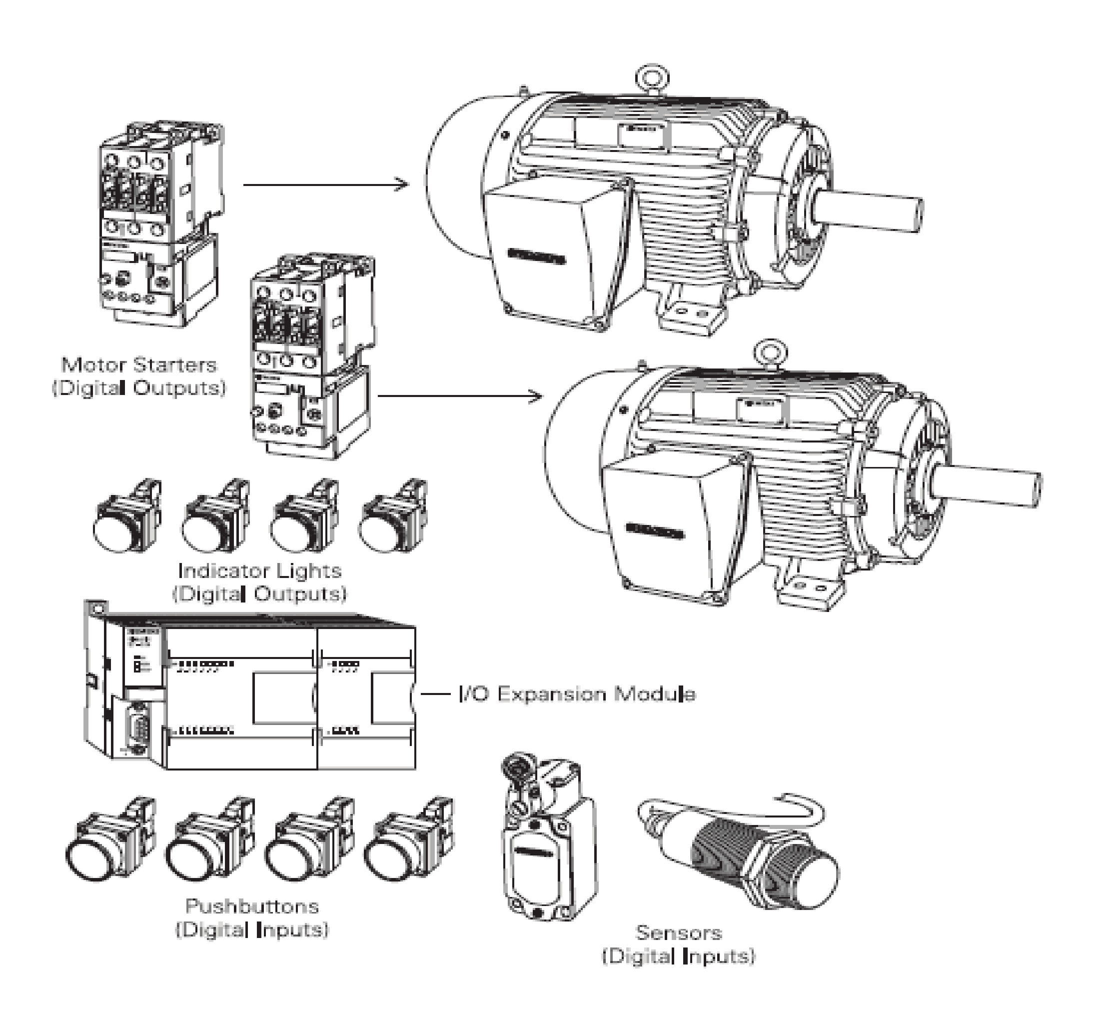


This control task can also be accomplished with a PLC.



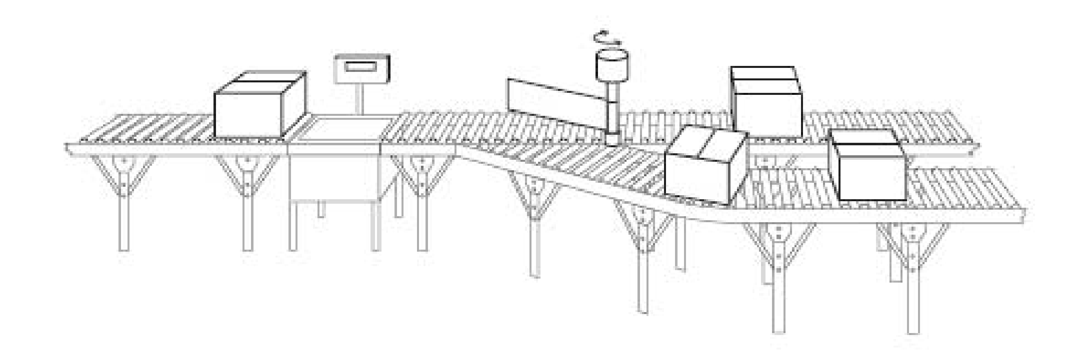
The PLC program can be expanded to accommodate many commercial and industrial applications. Additional Start/Stop pushbuttons and indicator lights can be added for remote operation, or control of a second motor starter and motor.

Over travel limit switches can be added along with proximity switches for sensing object position. In addition, expansion modules can be added to further increase the I/O capability. The applications are only limited by the number of I/Os and amount of memory available on the PLC.



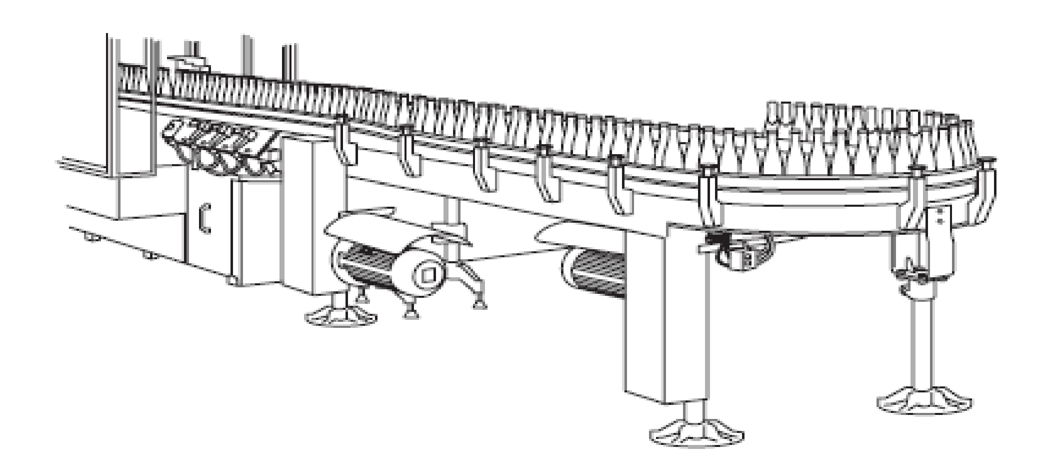
٢ Conveyor System (Example ٢) : -

The example application can be expanded to include a conveyor system with a gate to direct packages of varying weight. As packages move along the conveyor they are weighed. A package that weighs at or greater than a specified value is routed along one conveyor path. A package that weighs less than a specified value is routed along another conveyor path, where it will later be inspected for missing contents.



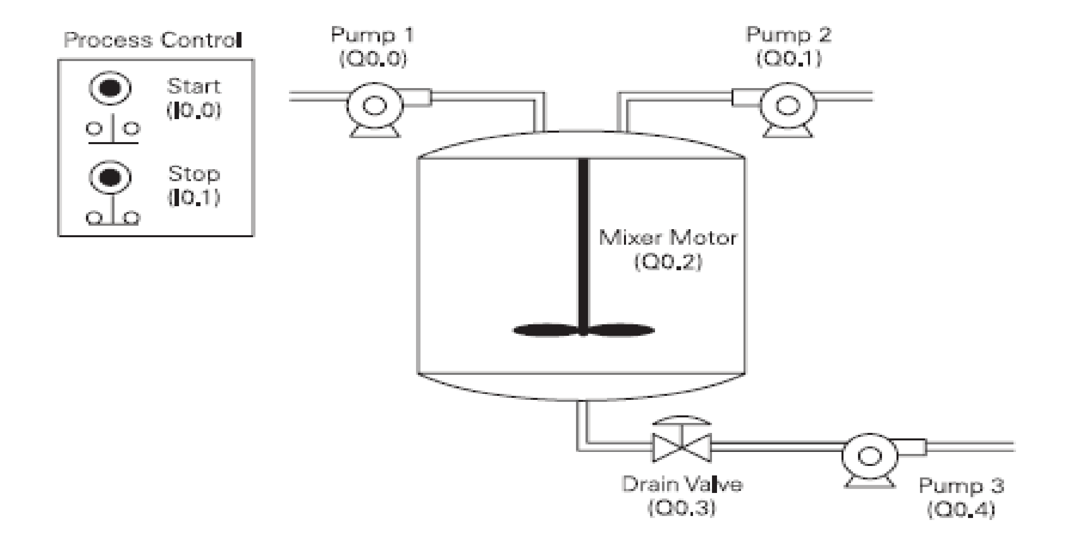
٣ A bottling machine (example ٣) : -

In this example we use a counter to count bottles into groups of six for packaging.



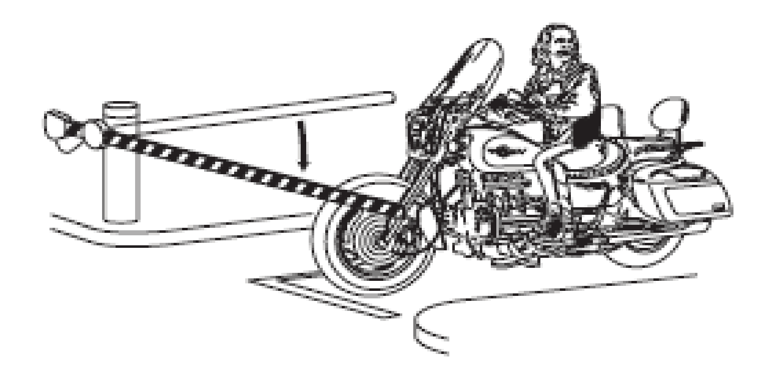
٤ Mixer (Example ٤) : -

In the following example a tank will be filled with two chemicals, mixed, and then drained. When the Start Button is pressed at input I٠.٠, the program starts pump ١ controlled by output Q٠.٠. Pump ١ runs for ٥ seconds, filling the tank with the first chemical, then shuts off. The program then starts pump ٢, controlled by output Q٠.١. Pump ٢ runs for ٣ seconds filling the tank with the second chemical. After ٣ seconds pump ٢ shuts off. The program starts the mixer motor, connected to output Q٠.٢ and mixes the two chemicals for ٦٠ seconds. The program then opens the drain valve controlled by output Q٠.٣, and starts pump ٣ controlled by output Q٠.٤. Pump ٣ shuts off after ٨ seconds and the process stops. A manual Stop switch is also provided at input I٠.١.



٥ **Parking (Example ٥) : -**

A counter might be used to keep track of the number of vehicles in a parking lot. As vehicles enter the lot through an entrance gate, the counter counts up. As vehicles exit the lot through an exit gate, the counter counts down. When the lot is full a sign at the entrance gate turns on indicating the lot is full.



٦- Appendix : -

Electrical Circuit Symbol : -

Circuit symbols are used in circuit diagrams which show how a circuit is connected together. The actual layout of the components is usually quite different from the circuit diagram. To build a circuit you need a different diagram showing the layout of the parts on strip board or printed circuit board.

Wires and connections		
Component	**Circuit Symbol**	**Function of Component**
Wire		To pass current very easily from one part of a circuit to another.
Wires joined		A 'blob' should be drawn where wires are connected (joined), but it is sometimes omitted. Wires connected at 'crossroads' should be staggered slightly to form two T-junctions, as shown on the right.
Wires not joined		In complex diagrams it is often necessary to draw wires crossing even though they are not connected. I prefer the 'hump' symbol shown on the right because the simple crossing on the left may be misread as a join where you have forgotten to add a 'blob'!

Power Supplies		
Component	**Circuit Symbol**	**Function of Component**
Cell		Supplies electrical energy. The larger terminal (on the left) is positive (+). A single cell is often called a battery, but strictly a battery is two or more cells joined together.
Battery		Supplies electrical energy. A battery is more than one cell. The larger terminal (on the left) is positive (+).
DC supply	+ −	Supplies electrical energy. DC = Direct Current, always flowing in one direction.
AC supply	~	Supplies electrical energy. AC = Alternating Current, continually changing direction.
Fuse		A safety device which will 'blow' (melt) if the current flowing through it exceeds a specified value.

Transformer		Two coils of wire linked by an iron core. Transformers are used to step up (increase) and step down (decrease) AC voltages. Energy is transferred between the coils by the magnetic field in the core. There is no electrical connection between the coils.
Earth (Ground)		A connection to earth. For many electronic circuits this is the ٠V (zero volts) of the power supply, but for mains electricity and some radio circuits it really means the earth. It is also known as ground.

Output Devices: Lamps, Heater, Motor, etc.

Component	Circuit Symbol	Function of Component
Lamp (lighting)		A transducer which converts electrical energy to light. This symbol is used for a lamp providing illumination, for example a car headlamp or torch bulb.
Lamp (indicator)		A transducer which converts electrical energy to light. This symbol is used for a lamp which is an indicator, for example a warning light on a car dashboard.
Heater		A transducer which converts electrical energy to heat.
Motor	M	A transducer which converts electrical energy to kinetic energy (motion).
Bell		A transducer which converts electrical energy to sound.
Buzzer		A transducer which converts electrical energy to sound.
Inductor (Coil, Solenoid)		A coil of wire which creates a magnetic field when current passes through it. It may have an iron core inside the coil. It can be used as a transducer converting electrical energy to mechanical energy by pulling on something.

Switches

Component	Circuit Symbol	Function of Component
Push Switch (push-to-make)		A push switch allows current to flow only when the button is pressed. This is the switch used to operate a doorbell.
Push-to-Break Switch		This type of push switch is normally closed (on), it is open (off) only when the button is pressed.

On-Off Switch (SPST)		SPST = Single Pole, Single Throw. An on-off switch allows current to flow only when it is in the closed (on) position.
٢-way Switch (SPDT)		SPDT = Single Pole, Double Throw. A ٢-way changeover switch directs the flow of current to one of two routes according to its position. Some SPDT switches have a central off position and are described as 'on-off-on'.
Dual On-Off Switch (DPST)		DPST = Double Pole, Single Throw. A dual on-off switch which is often used to switch mains electricity because it can isolate both the live and neutral connections.
Reversing Switch (DPDT)		DPDT = Double Pole, Double Throw. This switch can be wired up as a reversing switch for a motor. Some DPDT switches have a central off position.
Relay	NO COM NC	An electrically operated switch, for example a ٩V battery circuit connected to the coil can switch a ٢٣٠V AC mains circuit. NO = Normally Open, COM = Common, NC = Normally Closed.

Resistors

Component	Circuit Symbol	Function of Component
Resistor		A resistor restricts the flow of current, for example to limit the current passing through an LED. A resistor is used with a capacitor in a timing circuit.
Variable Resistor (Rheostat)		This type of variable resistor with ٢ contacts (a rheostat) is usually used to control current. Examples include: adjusting lamp brightness, adjusting motor speed, and adjusting the rate of flow of charge into a capacitor in a timing circuit.
Variable Resistor (Potentiometer)		This type of variable resistor with ٣ contacts (a potentiometer) is usually used to control voltage. It can be used like this as a transducer converting position (angle of the control spindle) to an electrical signal.
Variable Resistor (Preset)		This type of variable resistor (a preset) is operated with a small screwdriver or similar tool. It is designed to be set when the circuit is made and then left without further adjustment. Presets are cheaper than normal variable resistors so they are often used in projects to reduce the cost.

Capacitors

Component	Circuit Symbol	Function of Component

Capacitor		A capacitor stores electric charge. A capacitor is used with a resistor in a timing circuit. It can also be used as a filter, to block DC signals but pass AC signals.
Capacitor, polarised	+	A capacitor stores electric charge. This type must be connected the correct way round. A capacitor is used with a resistor in a timing circuit. It can also be used as a filter, to block DC signals but pass AC signals.
Variable Capacitor		A variable capacitor is used in a radio tuner.
Trimmer Capacitor		This type of variable capacitor (a trimmer) is operated with a small screwdriver or similar tool. It is designed to be set when the circuit is made and then left without further adjustment.

Diodes

Component	Circuit Symbol	Function of Component
Diode		A device which only allows current to flow in one direction.
LED Light Emitting Diode		A transducer which converts electrical energy to light.
Zener Diode		A special diode which is used to maintain a fixed voltage across its terminals.
Photodiode		A light-sensitive diode.

Transistors

Component	Circuit Symbol	Function of Component
Transistor NPN		A transistor amplifies current. It can be used with other components to make an amplifier or switching circuit.
Transistor PNP		A transistor amplifies current. It can be used with other components to make an amplifier or switching circuit.
Phototransistor		A light-sensitive transistor.

Audio and Radio Devices

Component	Circuit Symbol	Function of Component
Microphone		A transducer which converts sound to electrical energy.
Earphone		A transducer which converts electrical energy to sound.
Loudspeaker		A transducer which converts electrical energy to sound.
Piezo Transducer		A transducer which converts electrical energy to sound.
Amplifier (general symbol)		An amplifier circuit with one input. Really it is a block diagram symbol because it represents a circuit rather than just one component.
Aerial (Antenna)		A device which is designed to receive or transmit radio signals. It is also known as an antenna.

Meters and Oscilloscope

Component	Circuit Symbol	Function of Component
Voltmeter	V	A voltmeter is used to measure voltage. The proper name for voltage is 'potential difference', but most people prefer to say voltage!
Ammeter	A	An ammeter is used to measure current.
Galvanometer		A galvanometer is a very sensitive meter which is used to measure tiny currents, usually ١mA or less.
Ohmmeter	Ω	An ohmmeter is used to measure resistance. Most multimeters have an ohmmeter setting.
Oscilloscope		An oscilloscope is used to display the shape of electrical signals and it can be used to measure their voltage and time period.

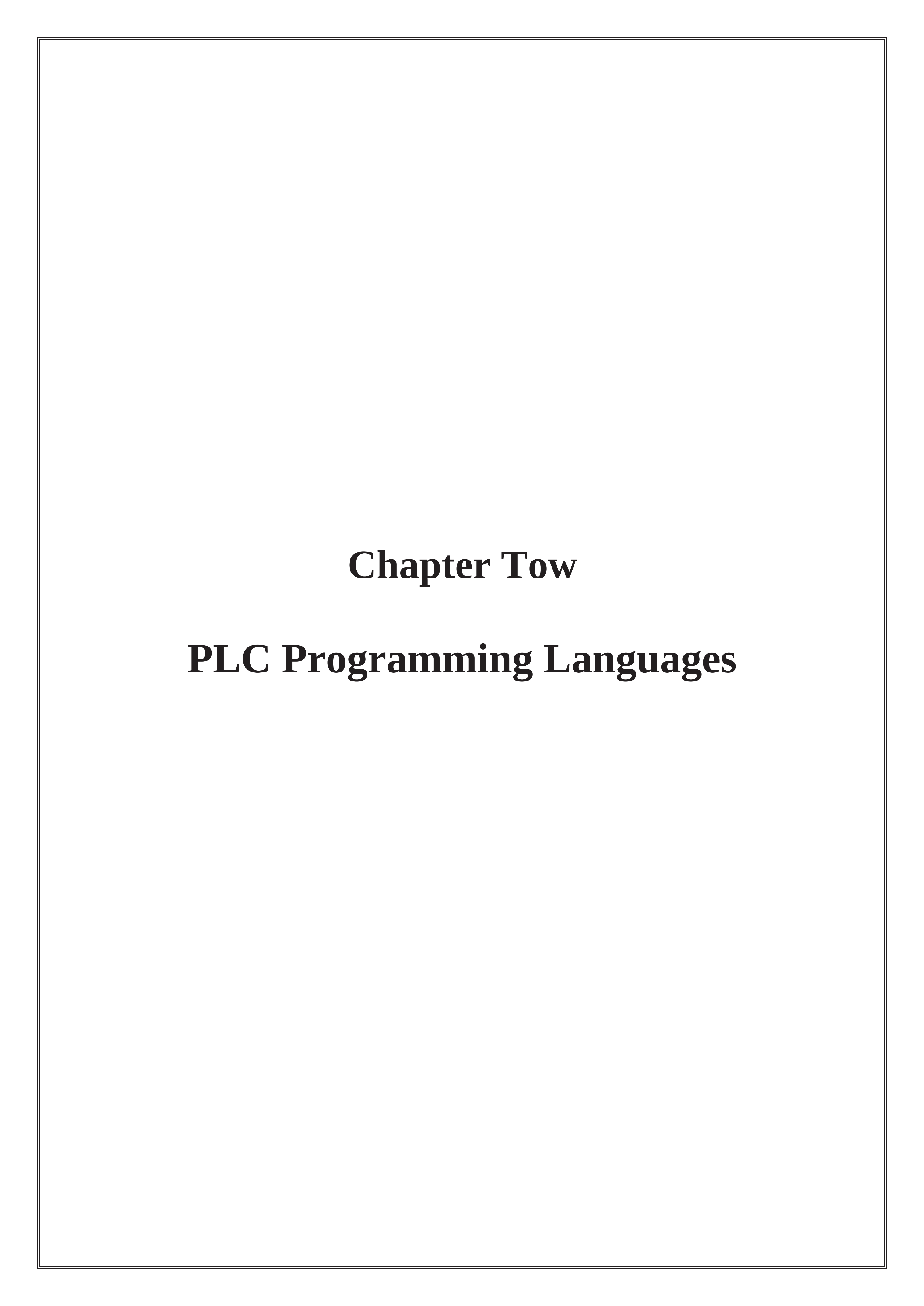
Sensors (input devices)

Component	Circuit Symbol	Function of Component
LDR		A transducer which converts brightness (light) to resistance (an electrical property). LDR = Light Dependent Resistor
Thermostat		A transducer which converts temperature (heat) to resistance (an electrical property).

Logic Gates

Logic gates process signals which represent **true** (١, high, +Vs, on) or **false** (٠, low, ٠V, off).
For more information please see the Logic Gates page.
There are two sets of symbols: traditional and IEC (International Electrotechnical Commission).

Gate Type	Traditional Symbol	IEC Symbol	Function of Gate
NOT		=1	A NOT gate can only have one input. The 'o' on the output means 'not'. The output of a NOT gate is the inverse (opposite) of its input, so the output is true when the input is false. A NOT gate is also called an inverter.
AND		&	An AND gate can have two or more inputs. The output of an AND gate is true when all its inputs are true.
NAND		&	A NAND gate can have two or more inputs. The 'o' on the output means 'not' showing that it is a Not AND gate. The output of a NAND gate is true unless all its inputs are true.
OR		≥1	An OR gate can have two or more inputs. The output of an OR gate is true when at least one of its inputs is true.
NOR		≥1	A NOR gate can have two or more inputs. The 'o' on the output means 'not' showing that it is a Not OR gate. The output of a NOR gate is true when none of its inputs are true.
EX-OR		=1	An EX-OR gate can only have two inputs. The output of an EX-OR gate is true when its inputs are different (one true, one false).
EX-NOR		=1	An EX-NOR gate can only have two inputs. The 'o' on the output means 'not' showing that it is a Not EX-OR gate. The output of an EX-NOR gate is true when its inputs are the same (both true or both false).



Chapter Tow

PLC Programming Languages

Types PLC Programming Languages : -

Instruction List (IL)

Series of instructions, each one must start on a new line.

One instruction = operator + one or more operations separated by commas.

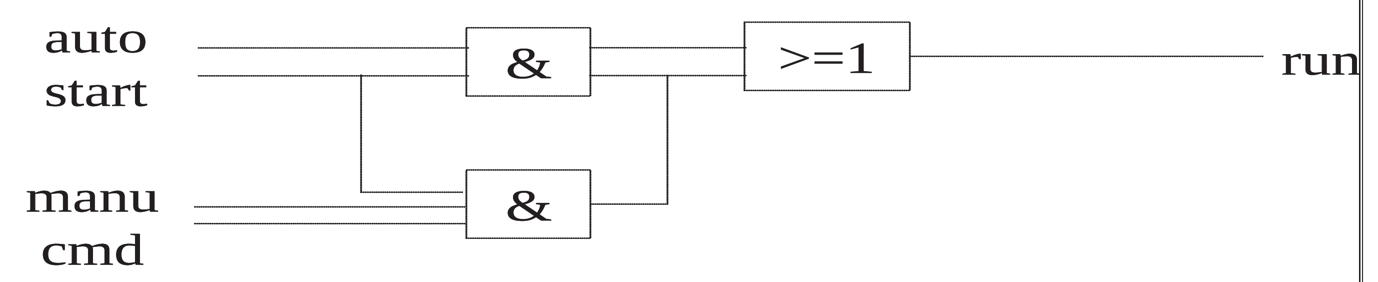
Function Blocks lunched using a special operator.

Label	Operator	Operation	Comment
Run:	LD	%IX١	(*pushbutton*)
	ANDN	%MX٥	
	ST	%QX٢	(*run*)

Function Block Diagram (FBD) : -

Representation of functions by blocks linked to each other.

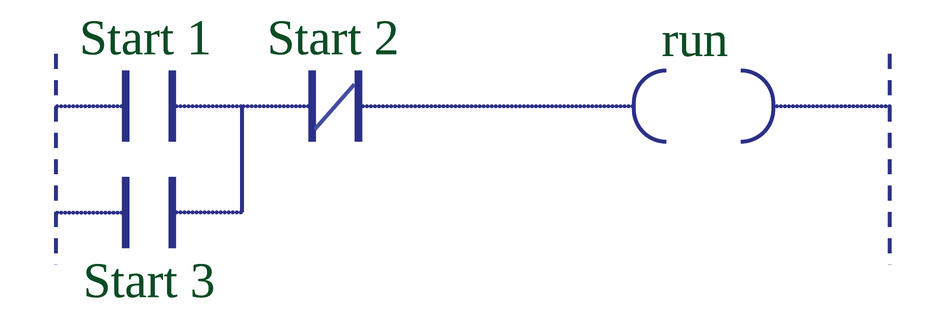
Network evaluation :from the O/P of a function block to the I/P of the connected function block.



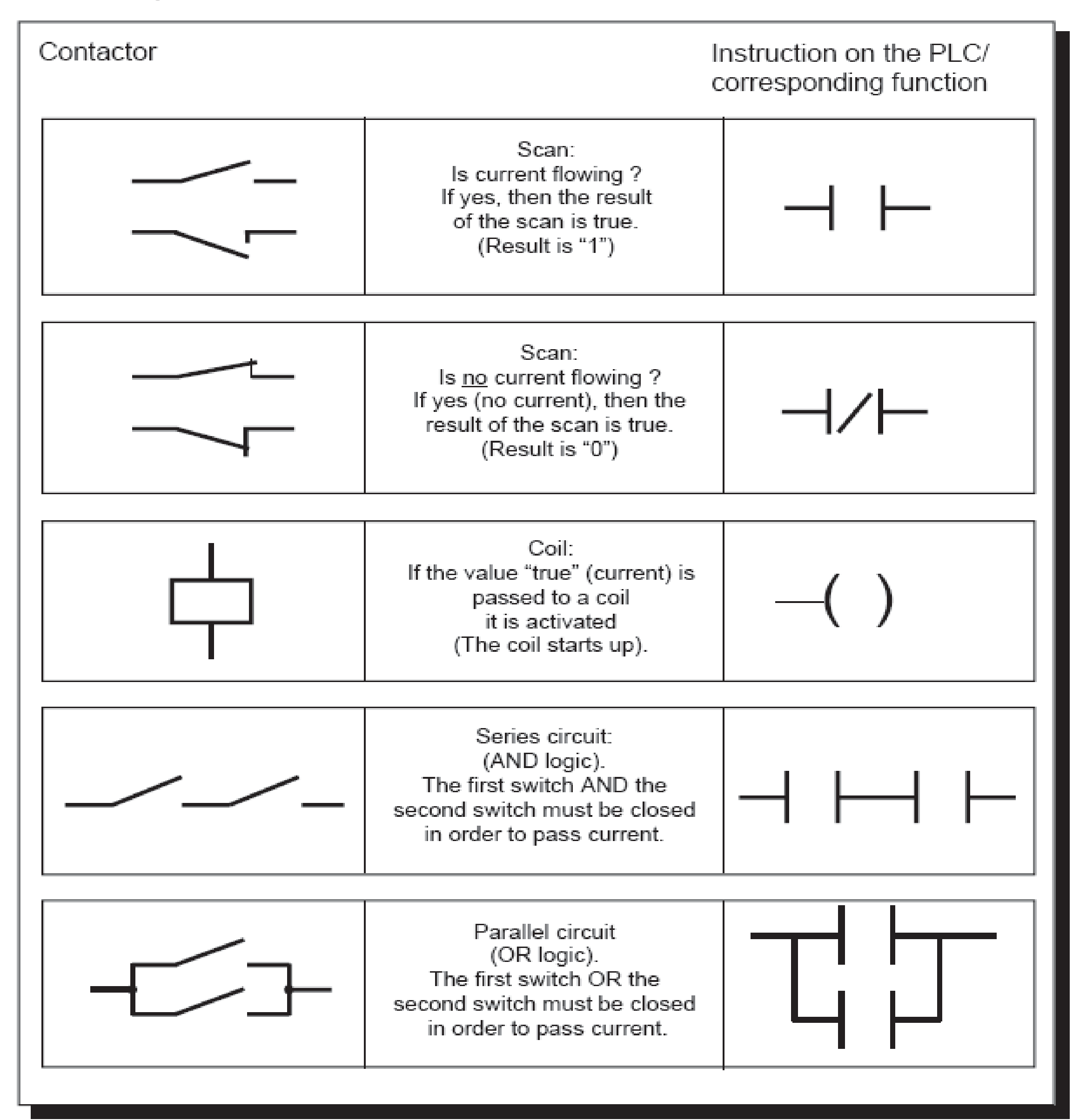
Ladder Diagram (LD) : -

Graphic elements organized in networks connected by power supply rails.

Elements used :contacts, coil, functions, function blocks control elements (jump, return, etc.)



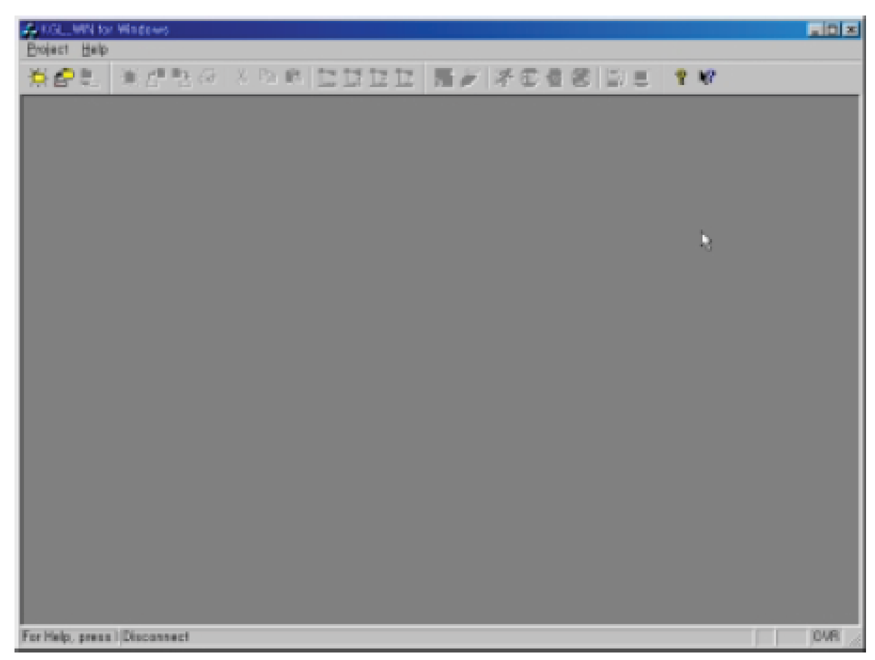
Ladder Logic Elements :



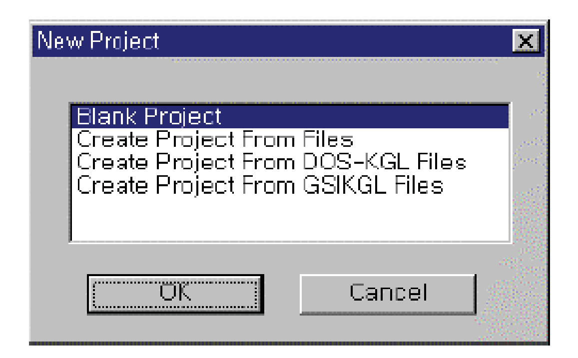
Contactor		Instruction on the PLC/ corresponding function
	Scan: Is current flowing ? If yes, then the result of the scan is true. (Result is "1")	
	Scan: Is no current flowing ? If yes (no current), then the result of the scan is true. (Result is "0")	
	Coil: If the value "true" (current) is passed to a coil it is activated (The coil starts up).	
	Series circuit: (AND logic). The first switch AND the second switch must be closed in order to pass current.	
	Parallel circuit (OR logic). The first switch OR the second switch must be closed in order to pass current.	

<u>Creating a Project :</u>

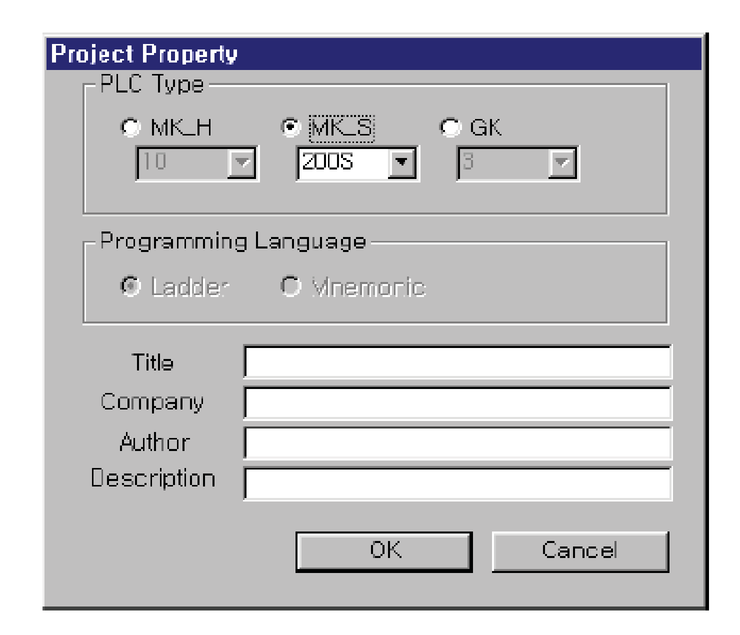
- **Double-click KGL_WE.exe file to run KGLWIN.**
- **The Start-up Screen will be shown as below.**



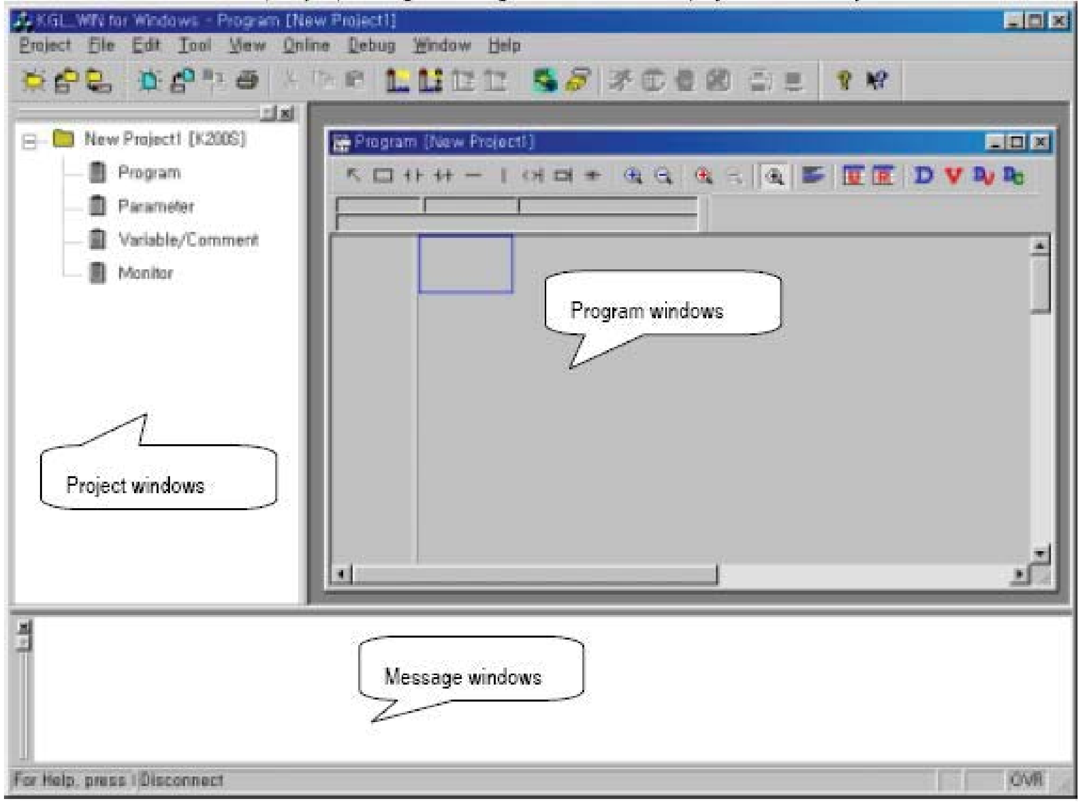
- To create a new project, select **Project- New Project...** ( ) in the Start-up Screen.
- Select **Blank Project** in the dialog box and click OK button.



In the following dialog box,
Type in PLC Type, Programming Language, Title, Company, Author and Description.



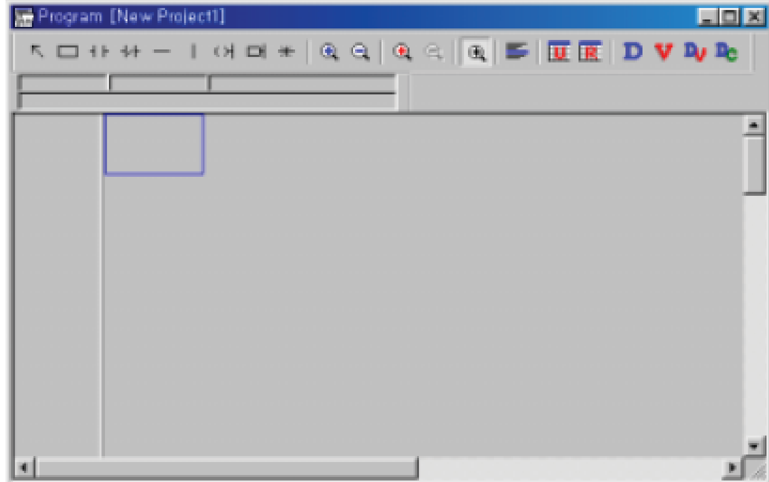
◆ Click OK button. Then, Project, Message and Program Windows are displayed automatically.



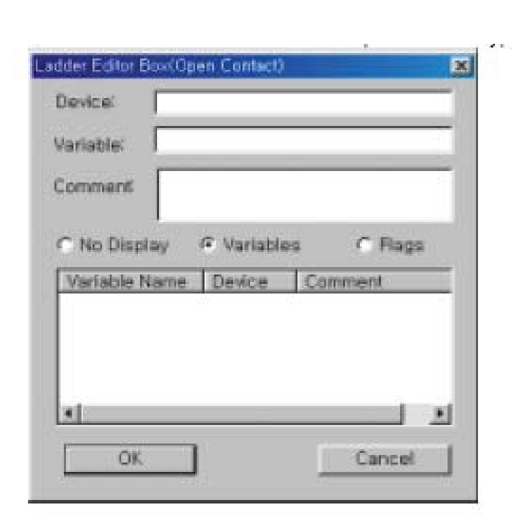
Creating a Program :

- **Creating a Ladder Program :**

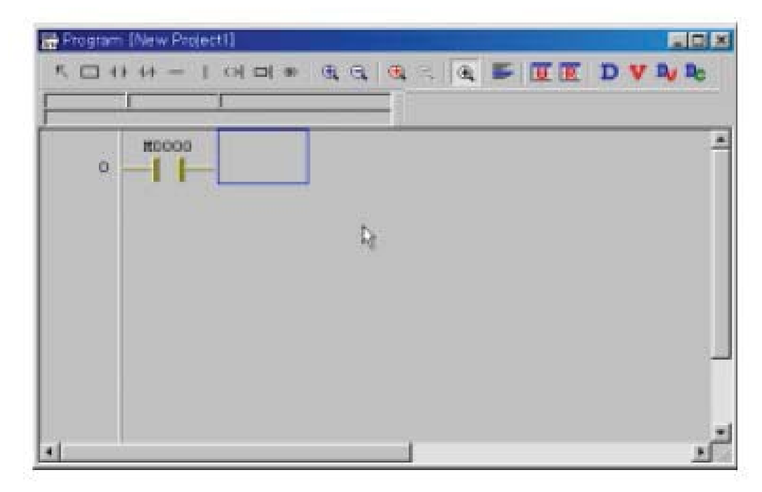

◆ This chapter describes creating a program in Ladder Window using the Tool Bar.



- **After selecting the Normally Open Contact (⊣⊢) icon in the Ladder Tool Bar, Move the cursor to the place to insert the contact.**
- **Click the left button of the mouse or press Enter key, then the contact input dialog box appears.**

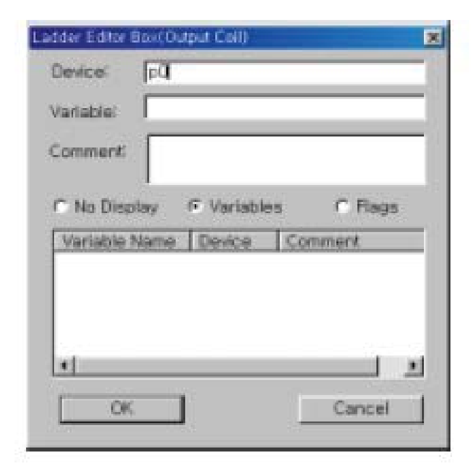


Type in the contact name(M٠٠٠٠) you want to insert and click OK button or press Enter key.

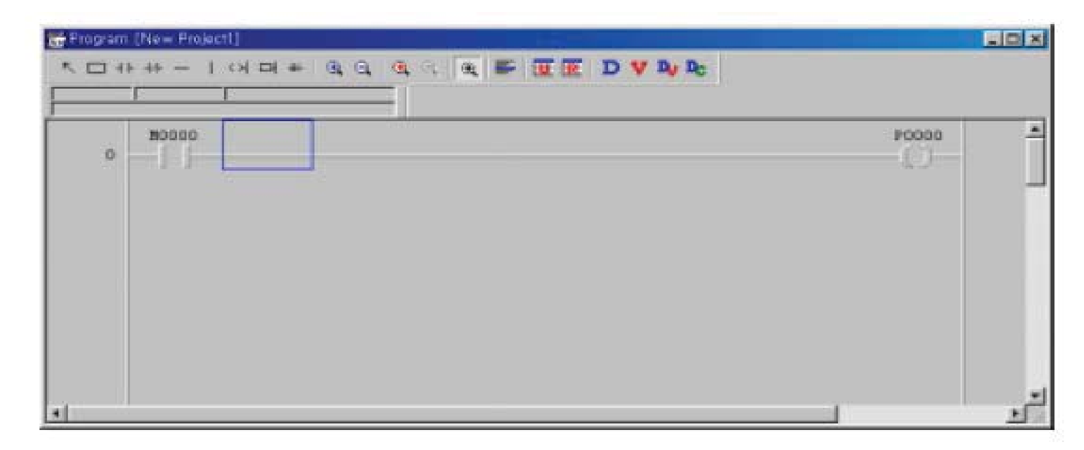


Select the Output Coil (⟨ ⟩) icon in the Ladder Tool Bar and move the cursor to the next column of M٠٠٠.

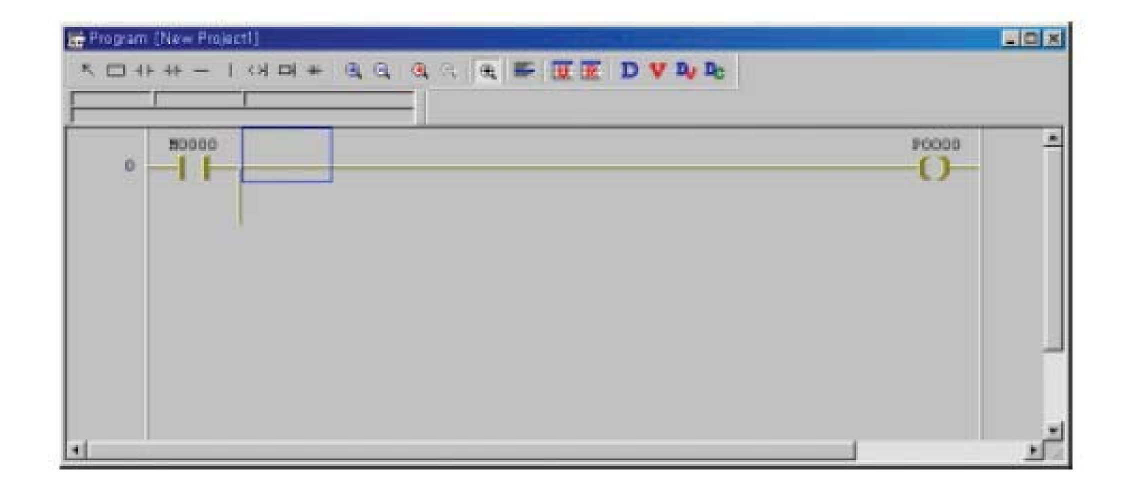
Click the mouse button or press Enter key.



Type in the Output Coil (P٠٠٠) and click OK button or press Enter key.



Select the Vertical line (|) icon in the Ladder Tool Box and click the mouse button on the cursor position.



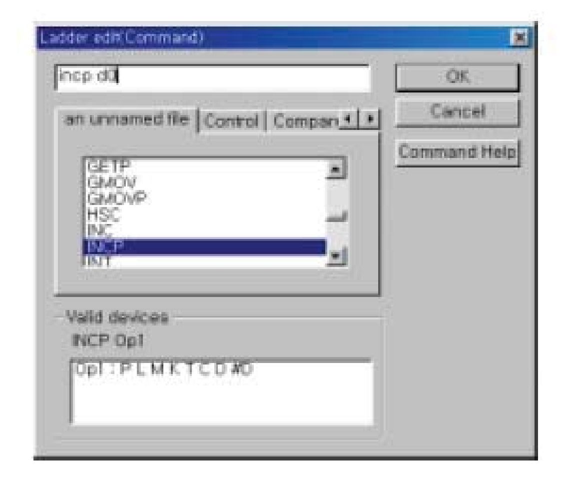
To delete Vertical line, select Vertical line and press del key or press back space key in above picture.

Select the Normally Closed Contact ( ) icon in the Ladder Tool Bar and Move the cursor to the place to insert the contact. Click the mouse or press Enter key to open the input dialog box for the contact input.



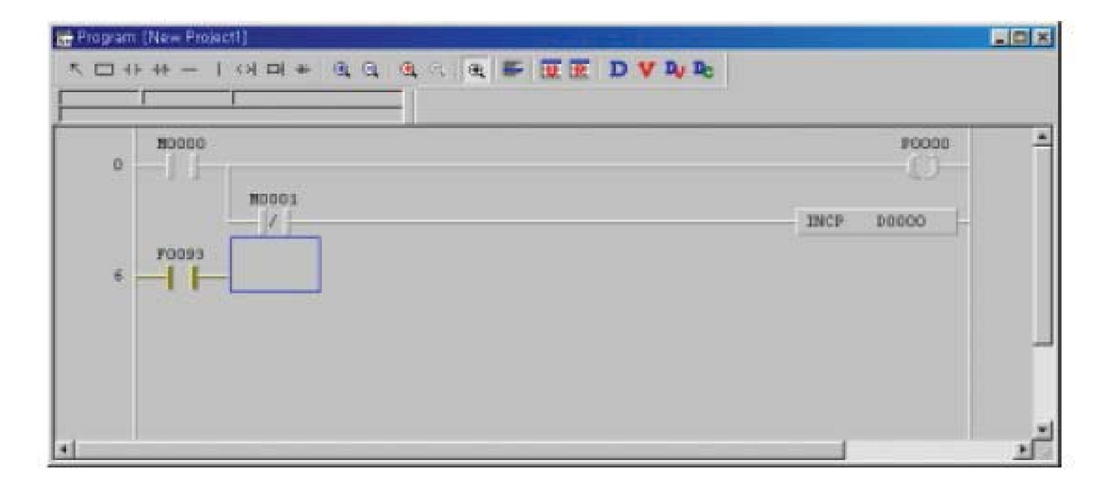
Type in an input contact that you want and click OK button or press Enter key.

- **After selecting the Applied Instruction icon( ) in Ladder Tool Box, click the mouse or press Enter key.**
- **Type INCP D٠٠٠٠ in the Ladder Editor Box.**

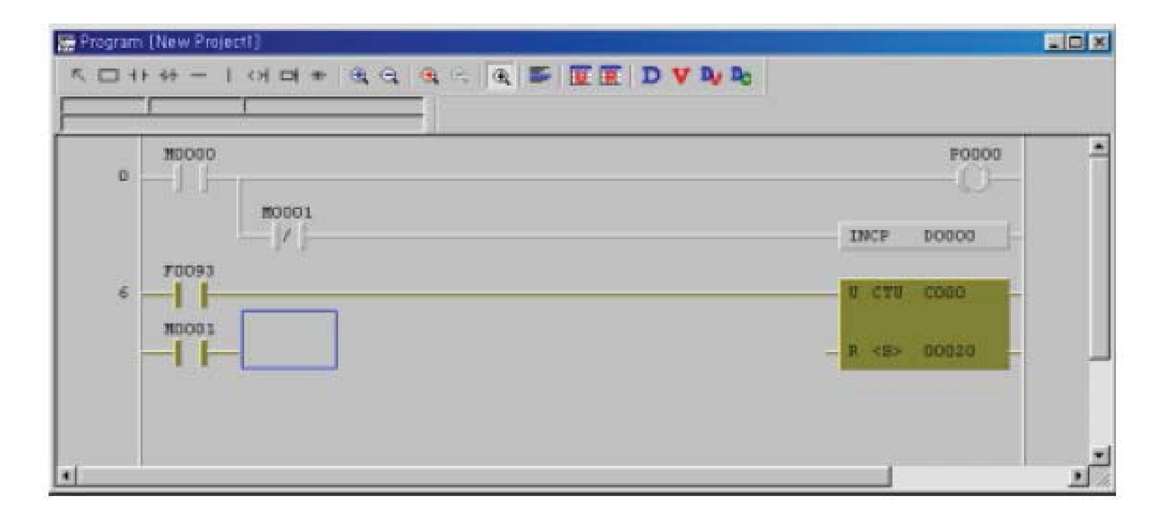


Select the Normally Open Contact (⊣ ⊢) icon in the Ladder Tool Bar and move the cursor to the next start line

- **Click the mouse or press Enter key.**
- **Type F٠٠٩٣ (System pulse clock for ١ second)**

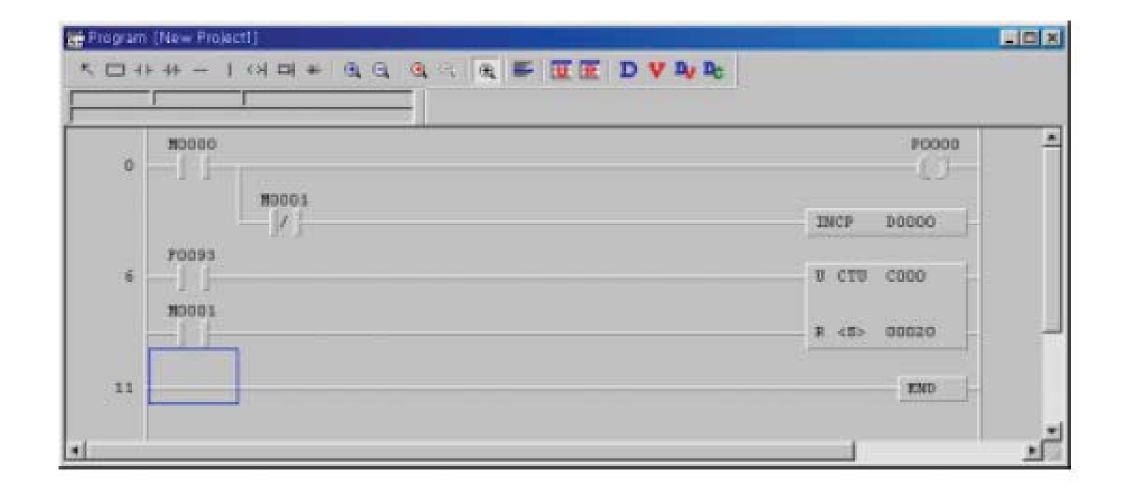


- **Select the Applied Instruction icon (⊔⊣) in the Ladder Tool Bar and type CTU C٠٠٠ ٢٠ .**
- **For the Reset input of the Counter, type M٠٠٠١ in the reset position after selecting the Normally Open Contact.**

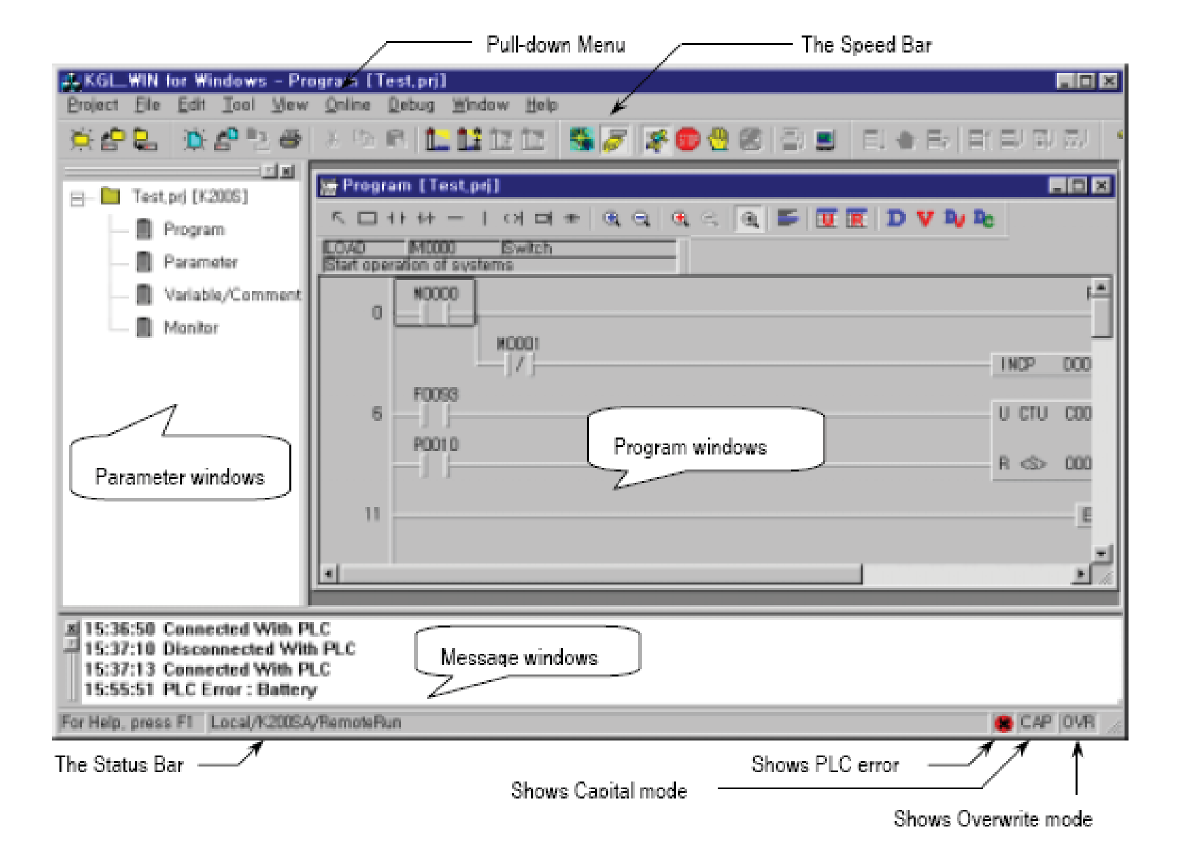


To complete the Ladder line, select the Horizontal Line ( ) icon in the Ladder Tool Box. And click the mouse to fill the spaces between two icons with the line.

To finish the program editing, insert END Instruction in the next line. Select the Applied Instruction icon ( ) to enter END instruction and press Enter key or click the mouse button.



The Screen Setup and Functions :



The Status Bar :

- Description Mode : Describes the Function of the Command or Menu.
- PLC Mode : Shows the Connection Status, PLC Type, Depth of Connection, PLC Operating Mode, View Mode, Monitoring Mode and more.

Menu :

♦ To perform an item (command) in Menu, you can select the Command in Pull-down menu or the icon in the Speed Bar.

♦ The Menu Bar provides two ways for you to access the Pull-down menus. Once the Pull-down menu appears, You can access the menu items in the same way by either using the mouse or pressing the underlined letter (Project…) In order to execute a command by the underlined character in the menu list, press the <ALT> key first, and then press the letter.

- **Project :**

Commands		Descriptions
New Project…		Create a new Project
Open Project…		Open an existing Project
Save Project		Save the active Project
Save Project As…		Save the active Project as a new Project.
Close Project		Close the current Project
Binary File	▸	Load / Make binary file
Save Item	▸	Save Program, Parameter and Variable/Comment as Files
Load Item	▸	Load Program, Parameter and Variable/Comment files in a Project
Options…		Set KGLWIN Options
Print…	Ctrl+P	Print the active document
Print Preview		Preview documents to print
Project Print		Print all the contents of the Project
Print Setup…		Setup the Printer Options.
Recent Project		Open the recent Project
Exit		Quit the application

- **File :**

Commands		Descriptions
New…	Ctrl+N	Create a new File
Open…	Ctrl+O	Open an existing File
Save	Ctrl+S	Save the active File
Save As…		Save the active File as a new File
Recent File		Open the recent Files

- **Edit :**

Commands		Descriptions
Cut	Ctrl+X	Remove the selected block and send it to the Windows clipboard
Copy	Ctrl+C	Copy the selected block to the Windows clipboard without affecting it
Paste	Ctrl+V	Retrieve it from the Windows clipboard
Delete	Ctrl+Delt	Delete the selected block
Insert contact mode	Insert	Change edit mode into insert mode
Insert Line	Ctrl+M	Insert a line at a caret position
Delete Line	Ctrl+U	Delete a line at a caret position
Edit Rung Comment		Edit a Comment at a specified rung
Block Selection…		Select a block using Step range
Optimize Program		Optimize the program
Find…	Ctrl+F	Find the specified string
Replace	Ctrl+H	Find the specified string and replace it a new device
Forward Again	Ctrl+F3	Search again forward
Backward Again	Ctrl+B	Search again backward.
Go to Step	Ctrl+G	Move the cursor to the step

- **Tool (Only available for Ladder Program) :**

Commands		Descriptions
Arrow		Go to Arrow mode
Range		Set the range for block
Normally Open Contact	F3	Select a Normally Open Contact
Normally Closed Contact	F4	Select a Normally Closed Contact
Horizontal Line	F5	Select a Horizontal Line
Vertical Line	F6	Select a Vertical Line
Output Coil	F9	Select a Output Coil
Applied Instruction	F10	Select an Applied Instruction
NOT Instruction	N	Select a NOT Instruction

- **View :**

Commands		Descriptions
Mnemonic/Ladder	Ctrl+Space	Change the program to Ladder or Mnemonic Mode
Device Name		Display the Device Name
Variable Name		Display the Variable Name.
Device+Variable Name		Display the Device and Variable Name
Device+Comment		Displays the Device and Variable Name
Used Device		Display the Used Device
Device Reference		Show the Device Reference
Check Program...		Check the errors in the Project
Change No of Column	▸	Change the number of the input contact in one line
Zoom In/Out	▸	Zooming

- **Online :**

Commands		Descriptions
Connect+Download+Run +Monitor Start	Ctrl+R	Execute Transfer, Run and Monitor Functions simultaneously
Connect		Connect to PLC
Disconnect		Disconnect from PLC
Monitor Start		Start Monitoring
Mode Change	Run	Change PLC Mode to Run
	Stop	Change PLC Mode to Stop
	Pause	Change PLC Mode to Pause
	Debug	Change PLC Mode to Debug
Read Information	System Monitor	Monitor ON/OFF status of I/O modules
	PLC Information...	Show PLC Type, PLC Version, Memory Pack and Scan Time
	I/O Information...	Show type of modules and O/S version number of special modules
	Link Information...	Show the slot number of Network modules
	Mnet Information...	show the slot number of Mnet module
	HSLink Parameter ▸	Monitor High Speed Link Parameter
	Comm info...	Monitor built-in Cnet communication (K200S/K80S only)
Write Information	Set PLC Clock...	Set PLC Clock
	Change Password...	Change the Password in PLC
	Write Mnet Parameter...	Change the Parameter of Mnet
	FSM Emergency Output	Setup the device for an emergency output
Download...		Download Program/Parameter to PLC

Upload...		Read Program/Parameter from PLC
Verify...		Verify Program/Parameter with PLC
Clear	Data...	Clear Devices in PLC
	Program/Parameter	Clear the program/parameter in PLC
Flash Memory	Read	read the program/parameter from Flash Memory
	Write	write PLC program/parameter to Flash Memory
	Verify	verify the program/parameter with Flash Memory
EPROM	Type Selection...	Select EPROM Type
	Write	write PLC program/parameter to EPROM
	Read	read the program/parameter from EPROM
	Verify	verify the program/parameter with EPROM
	Check Blank	Check if EPROM is empty
Binary File	Load Binary File...	transfer the program/parameter in KGLWIN to EPROM
	Make Binary File...	receive the Binary File from EPROM

- **<u>Debug :</u>**

Commands	Descriptions
Trace Ctrl+T	Run only one Step
Go	Run until current Break point
Stop	Stop Debugging
Break Step...	Run to the specified Break Step
Break Scan...	Run to the specified Break Scan
Break Bit...	Run until the specified Bit is set
Break Word	Run until the specified Word data
Change Current I/O... Ctrl+I	Change Current I/O
Forced I/O Enable	Enable Forced I/O setting
Set Forced I/O...	Set Forced I/O
Sampling Trace	Execute sampling Trace
Trigger	Execute Trigger

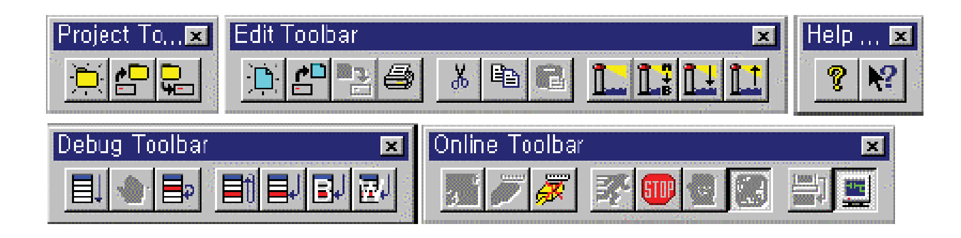
- **<u>Window :</u>**

Commands	Descriptions
New Window	Open a new window for the active program
Cascade	Cascade windows on the screen
Tile Horizontally	Arrange window as non-overlapping tiles
Tile Vertically	Arrange window as non-overlapping tiles
Arrange Icons	Arrange icons at the bottom of the window
Message Window	Open/Close the message window

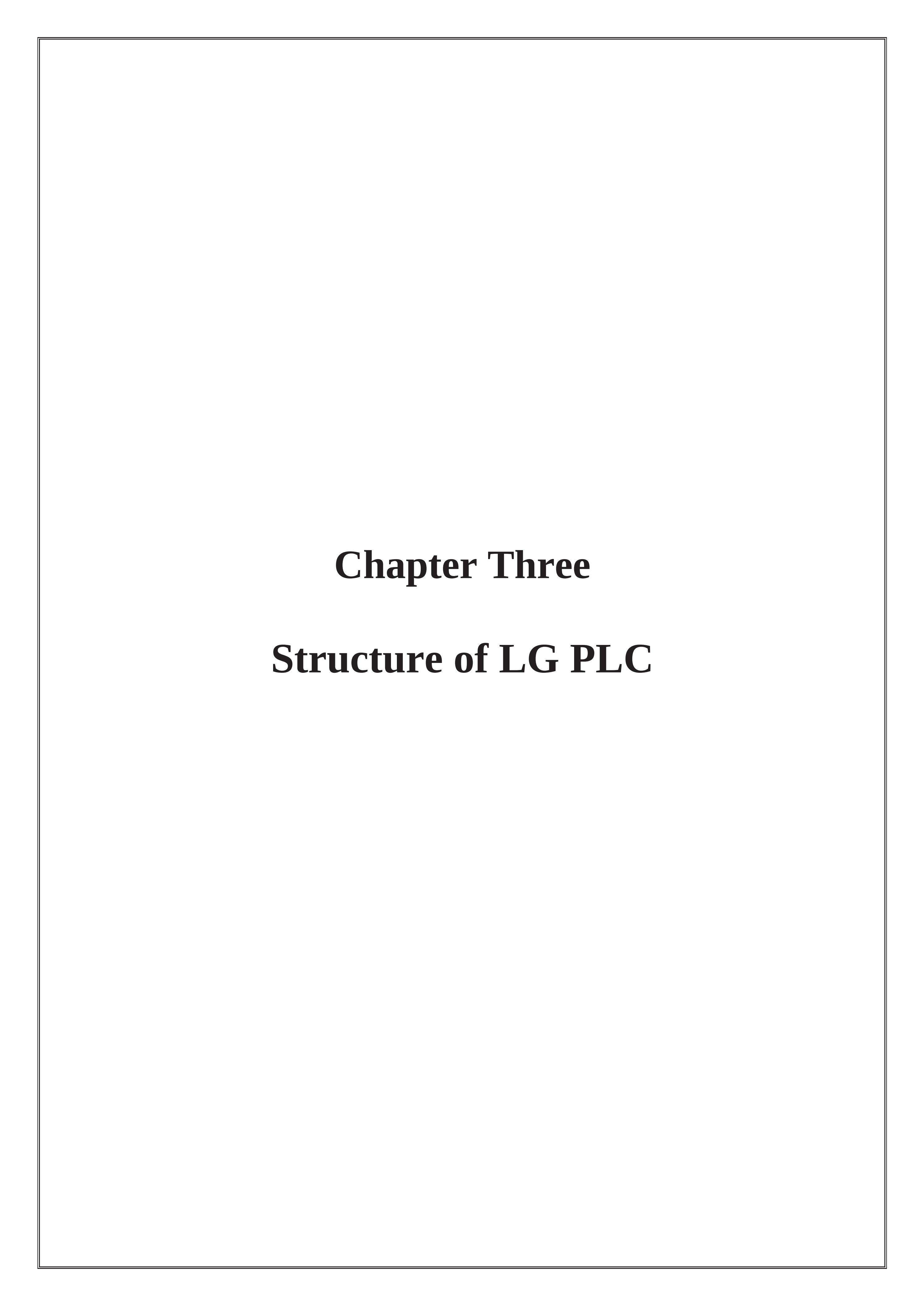
- ## Help :

Commands	Descriptions
KGLWIN Help	Display help topics of KGLWIN
Help in Help	Display detailed instructions about how to use
About KGLWIN...	Display general information of KGLWIN

Tool Bars :



Tools	Commands	Tools	Commands
	New Project		Connect
	Open Project		Disconnect
	Save Project		Download
	New File		Monitoring Mode
	Open File		Run
	Save File		Stop
	Print		Pause
	Cut		Debug
	Copy		Go
	Paste		Debug Stop
	Find		Trace
	Replace		Break Scan
	Forward		Break Step
	Backward		Break Bit
	Conncet+Dowload+Run+Monitor Start		Break Word



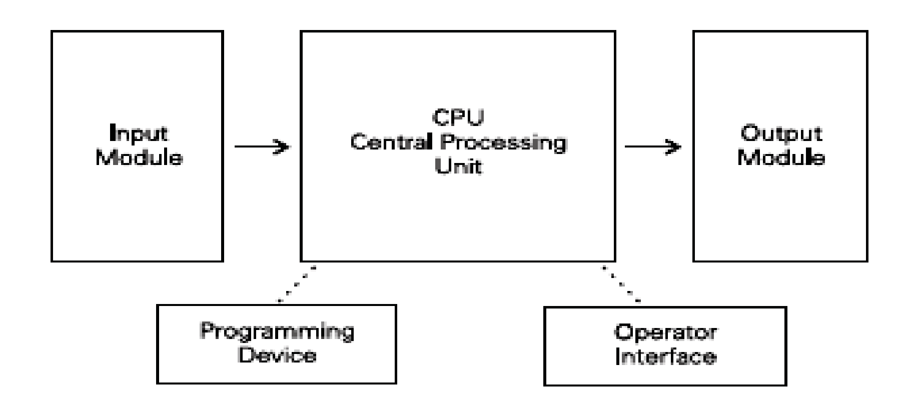
Chapter Three

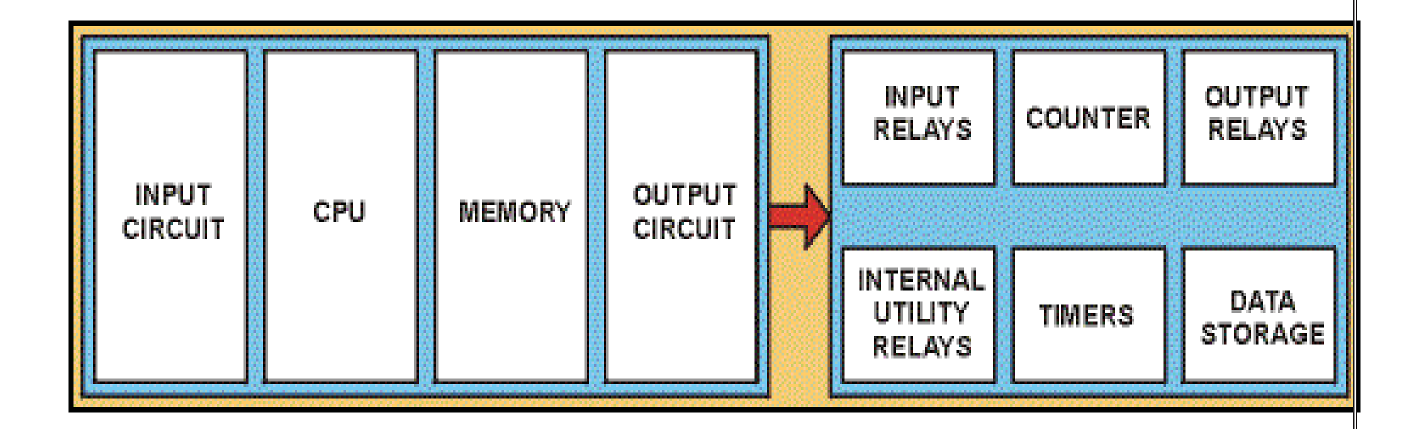
Structure of LG PLC

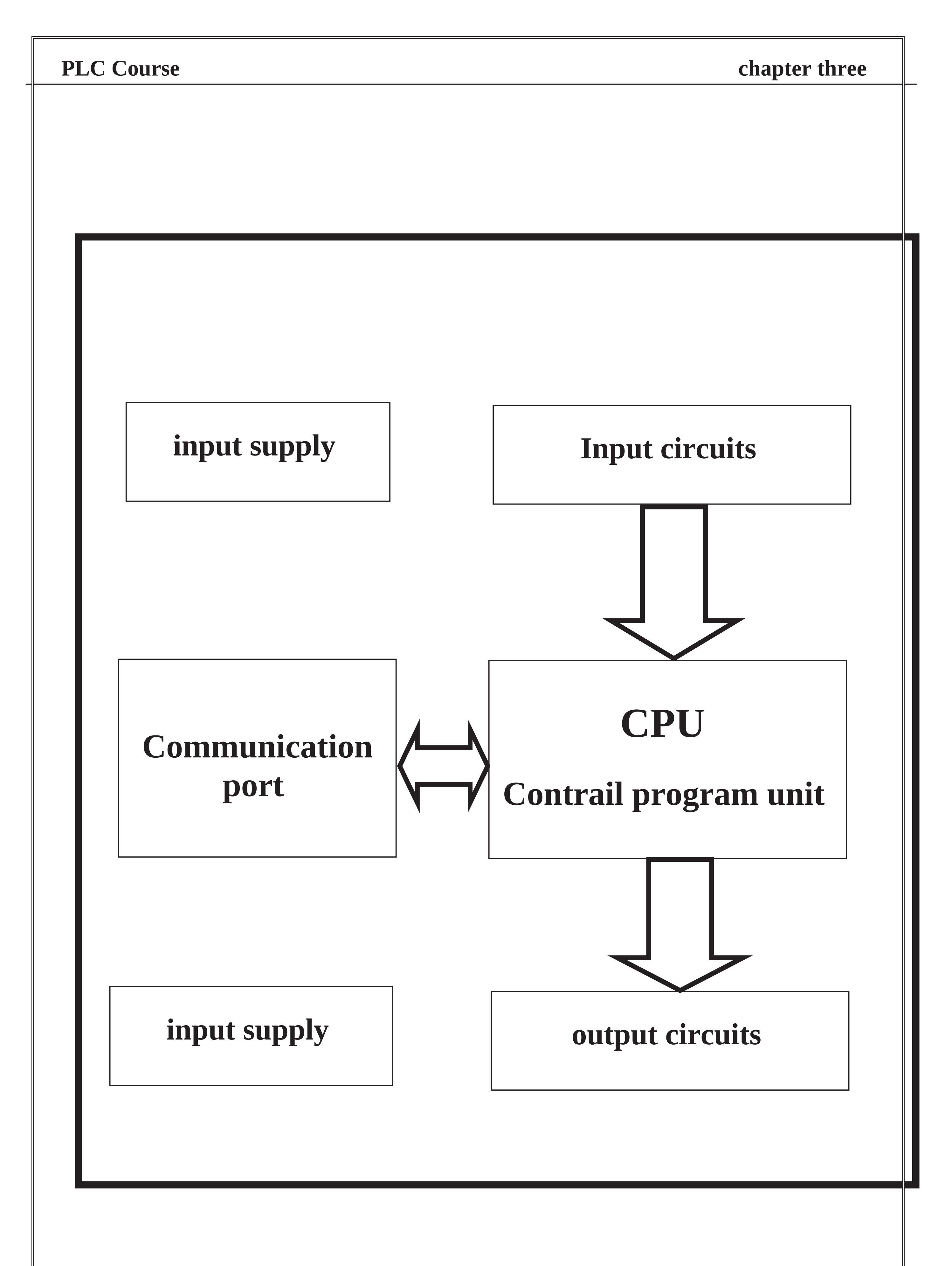
Structure of Plc

any type of plc having the same structure and it consist of : -

The PLC mainly consists of a CPU, memory areas, and appropriate circuits to receive input and output data. We can consider the PLC to be a box full of hundreds or thousands of separate relays, counters, timers and data storage locations. These counters, timers, etc. don't "physically" exist but instead are simulated and can be considered software counters, timers, etc. These internal relays are simulated through bit locations in Registers .






input supply
Input circuits
Communication port
CPU
Contrail program unit
input supply
output circuits

CPU

Like other computerized devices , there is a CPU in a PLC , the CPU which is the brain of the PLC is able to do the following operation : -

- **updating inputs and outputs this function allow PLC to read the status of the input terminal and energize or de energize output terminals .**

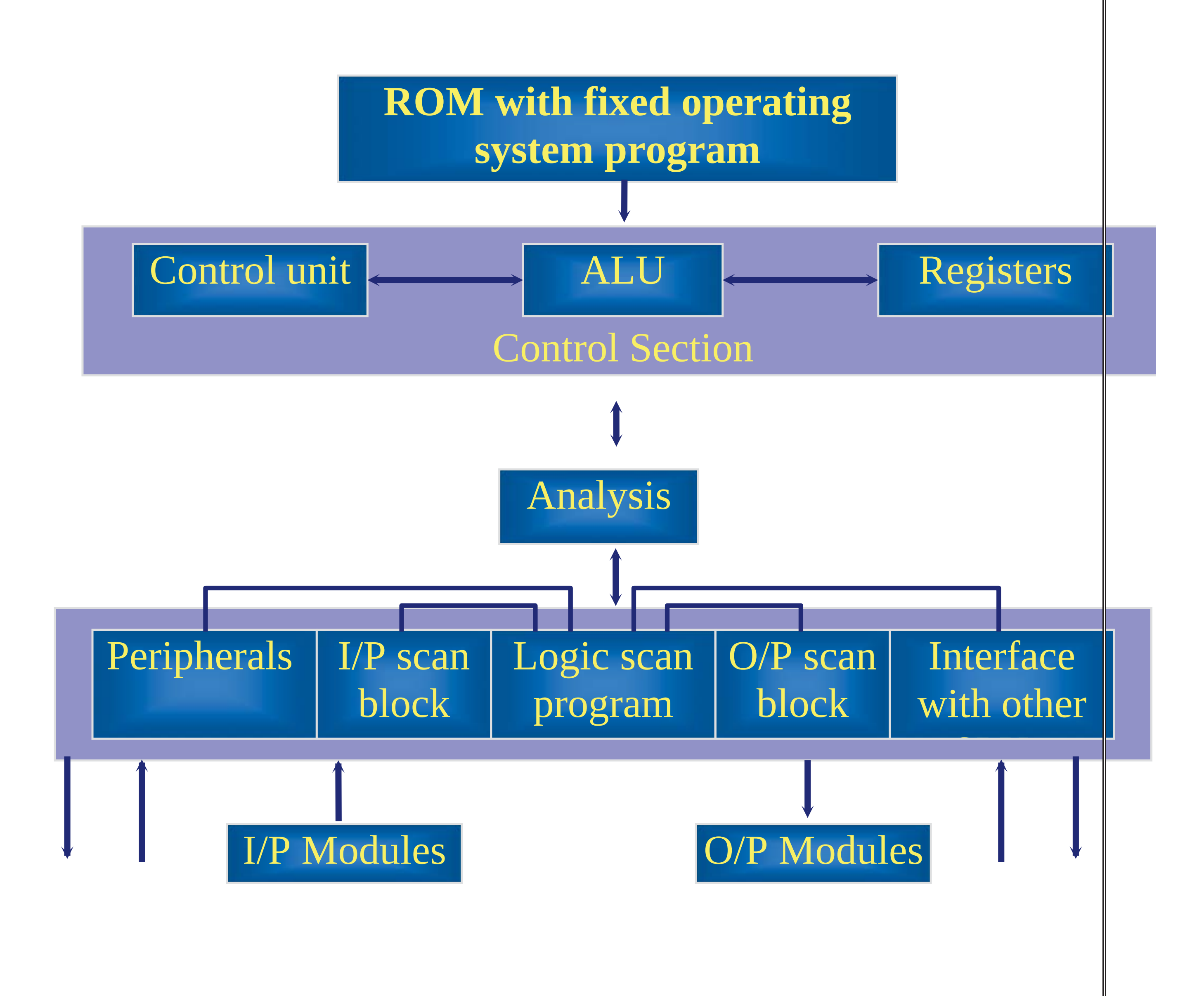
- **performing logic and arithmetic operation CPU conducts all the mathematical and logic operation involving in PLC .**

- **communication with memory . PLC's program and data are stored in memory . when a PLC is operating , its CPU reads or change the contents of memory location .**

- **scanning application program which is called ladder diagram this scanning allow PLC to execute the application program as specified by the programmer .**

- **The CPU controls and supervises all operation within PLC, carrying out programmed instructions stored in the memory.**
- **An internal communications highway or bus system carries information to and from CPU, memory and I/O units, under CPU control.**

- **The CPU is supplied with a clock frequency by a quartz crystal or RC oscillator with speed depending on the microprocessor type.**
- **The clock determines the operating speed of the PLC and provides timing/synchronization .**



A Memory of PLC : -

The operating system software is a group of supervisory programs that are loaded into the PLC's memory by the PLC manufacturer and stored there permanently.

Memory is the component to store information, programs and data, in a PLC. The process to put new information to a memory location is called *writing*. The process to retrieve information from a memory location is called *reading*.

The common types of memory used in PLCs are Read Only Memory (ROM) and Random Access Memory (RAM). The information stored in ROM can be read, but not written. The ROM is used to permanently store programs and data. For example, PLC's operating programs are stored in ROM.

A RAM location can be read or written. That means the information stored in a RAM location can be retrieved or altered. Ladder logic programs are stored in RAM. When a new ladder logic program is loaded into a PLC memory, the old one program that was stored in the same locations is erased.

The memory capacities of PLCs are different. Memory capacities are often expressed for RAM in terms of *kilo-bytes*. One byte is a group of 8 bits. One bit is a memory location that may store one binary number that is either 1 or 0. 1K memory means that there are 1024 bytes of RAM. 16K memory means there are 16 x 1024 =16384 bytes of RAM.

Power Supply : -

PLC is powered by commercial AC power lines . however many PLC component , such as CPU and memory use ٥ volt or other level of DC power . A PLC power supply converts AC power into DC power to support other component of PLC .

POWER SUPPLY : - **AC or DC according to application and system requirements and needs .**

١- To prevent the PLC from an improper operation caused by the external noise, place a insulation transformer and/or a noise filter as shown in the following figure.

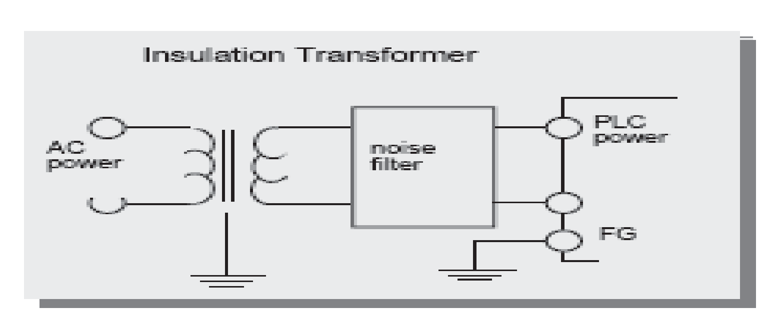
٢- Always install AC power cable and signal or data lines in separate ducts or bunches.

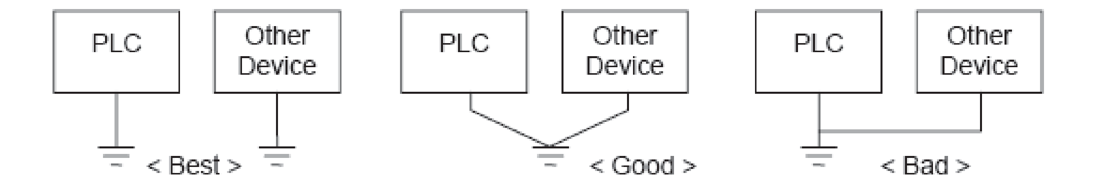
٣- The fuse in the DC power supply models, will be blown when the DC power is supplied in reverse polarity.

٤- Be careful to connect power source cable to the correct terminal. Internal device of PLC may be damaged by the improper lead connections.

٥- Supplying power beyond rated voltage / frequency may damaged internal devices of PLC.

٦- Grounding





Communication Port : -

A PLC needs a communication port as programming terminal and a copy of soft ware for operation . the programming terminal can be a dedicated terminal or a generic computer purchased any where . the programming terminal is used for programming the PLC and monitoring the PLC operation . it may also down load a ladder diagram (sending a program from programming terminal to PLC) . or up load a ladder logic program (sending program from PLC to program terminals) . PLC software provides the capability for programming terminal to program and to talk to PLC .

Input modules and out put modules : -

A PLC is a control device. It takes input information and makes decisions to energize or de-energize outputs. The decisions are made based on the statuses of input and outputs, and the ladder logic program that is running.

The input devices used with a PLC include pushbuttons, limit switches, relay contacts, photo sensors, proximity switches, temperature sensors, and etc. these input devices can be AC (alternating current) or DC (direct current). These input voltage can be high or low. The input signal can be digital or analog. To deal with different inputs , different input modules should be used. The input modules provide Interface between the input devices and the PLC CPU. Which uses only a low DC voltages. the input module's function is to convert the input signals to DC voltages that are acceptable by the CPU. The standard discrete input modules include ٢٤ V AC,٤٨ V AC, ١٢٠ V AC, ٢٢٠ V AC, ٢٤ V DC, ٤٨ V DC, ١٢٠ V DC,٢٢٠ V DC, and TTL level. .

The devices controlled by a PLC include relays, alarms, solenoids, fans, lights, and motor starters. These devices may use different levels of AC or DC voltages. It is an output module's job to convert the low DC voltage PLC control signals to the voltages that required by the controlled circuits or devices. The standard discrete output modules include ٢٤ V AC,٤٨ V ,AC, ١٢٠ V AC, ٢٢٠ V AC, ٢٤ V DC, ٤٨ V DC, ١٢٠ V DC,٢٢٠ V DC And TTL level .

There are two main type of input / output module : -

Discrete Inputs / outputs : -

Discrete is the most common class of input/output in a programmable controlled system . this type of interface module connect field devices that have two discrete states , such ON / OFF or OPEN / CLOSED to the processor .

In a discrete output module the output interface circuit switches relay or semiconductor types (transistor , triac , SCR , GTO , ……) .

Analog Inputs / Outputs : -

The analog I / O modules make it possible to monitor and control analog voltage and current , which are compatible with many sensors , motor drives , and process instruments . By using analog I / O , it is possible to measure or control most process variables with appropriate interfacing .

When selects I/O module for MASTER K PLC system, please refer the following

Instructions : -

١) The digital input module is classified as current sink input and current source input. The external wiring with input device is various according to the type of digital input module. You must select suitable digital input module type with considering of specification of input device.

٢) The maximum points that can be turn simultaneously on differ with each module. Before to select a digital I/O module, check the specification of module.

٣) When a very fast response time (less than a scan time) is required, select an interrupt module. However, only one interrupt module can be mounted on a system.

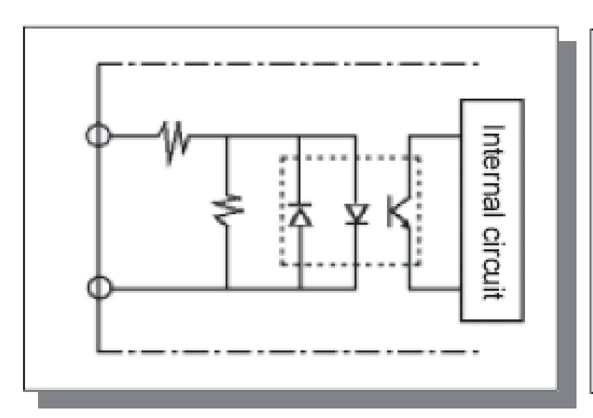
٤) The lifetime of relay is described as total on/off times (No load : ١٠ million times, With load : ٠.١ ~ ٣ million times). Therefore, if the frequency of on/off operation of relay is higher, the lifetime of relay is shorter. Please use transistor or SSR output module for high frequency operation.

٥) When a large and/or inductive load is connected directly to the output module, it may cause malfunction of the output module. It is highly recommended customers to connect an external relay or SSR between an output module and large inductive load for improved reliability and maintenance of PLC system.

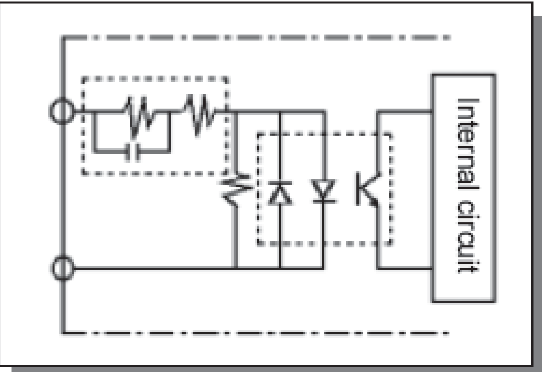
Terms	Definition	Remark
Sink input	Current flows in from the input switch to the input terminal of PLC when an input signal is turned on.	Z: Input impedance
Source input	Current flows out from the input terminal of PLC to the input switch when an input signal is turned on.	
Sink output	Current flows in from the external load to the output terminal of PLC when an output signal is turned on.	
Source output	Current flows out from the output terminal of PLC to the external load when an output signal is turned on.	

■ I/O circuit

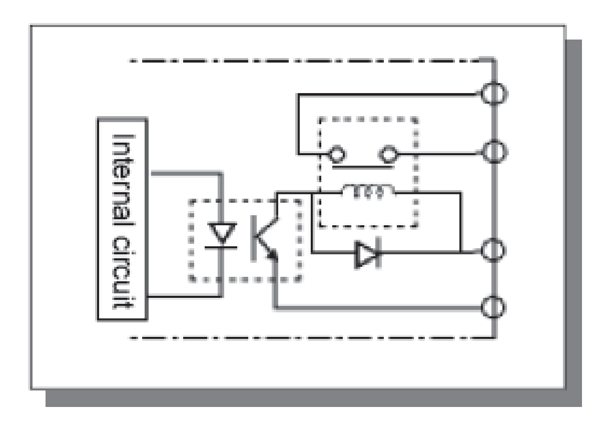
DC24V input



AC input

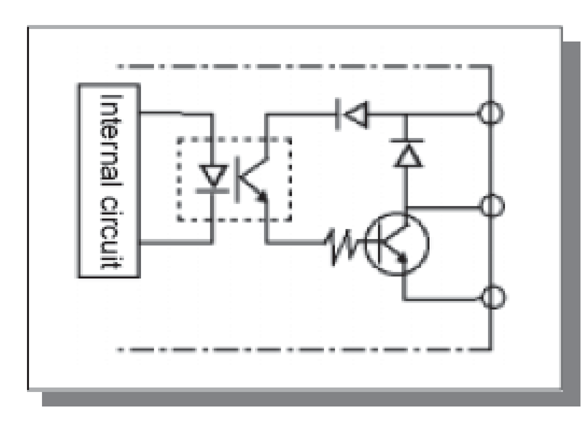


Relay output

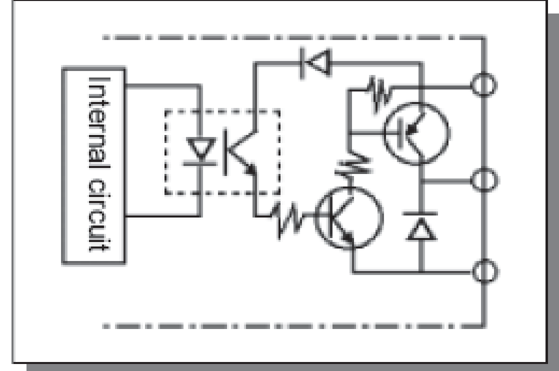



SSR output

Transistor (NPN) output

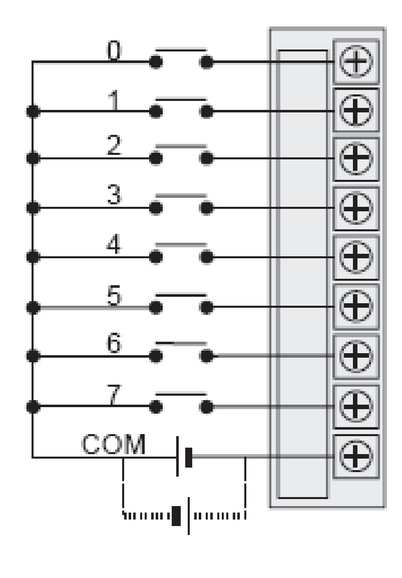


Transistor (PNP) output



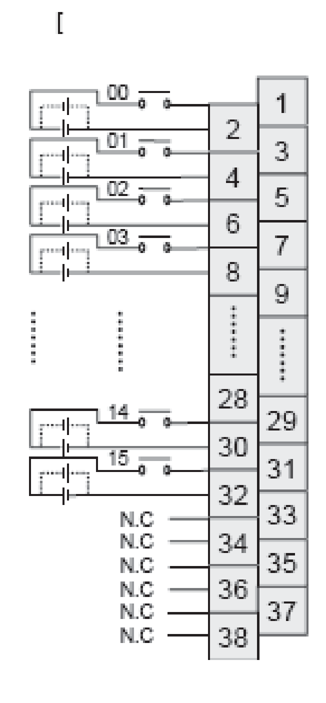
l 8 points 12/24VDC input module (source / sink type)

Specification \ Type		K200S	
		K3X-110S	
Input points		8 points	
Insulation method		Photo coupler insulation	
Rated input voltage		12VDC	24VDC
Rated input current		3 mA	7 mA
Operating input voltage		10.2 ~ 26.4 VDC (ripple : 5% or less)	
Max. simultaneously on		8 points (100%)	
On voltage / current		9.5 VDC / 3.5 mA	
Off voltage / current		5 VDC / 1.5 mA	
Input impedance		About 3.3kΩ	
Response time	Off → On	5 msec or less	
	On → Off	5 msec or less	
Common		8 points / 1 com	
Internal current consumption		40 mA	
Operation indicator		LED display	
External wiring		9 points terminal block connector (M3×6 screw)	
Weight		120 g	
Wiring diagram			



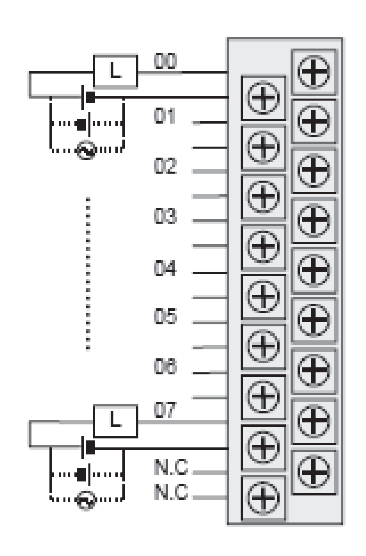
7 terrupt input module

Type / Specification		K300S	K1000S
		K4F-INTA	K7F-INTA
Input points		8 points	16 points
Insulation method		Photo coupler insulation	
Rated input voltage		24 VDC	
Rated input current		10 mA	
Operating input voltage		21.6 ~ 26.4 VDC	
Max. simultaneously on		1 points / 1COM (100%)	
On voltage / current		15 VAC / 6.5 mA	
Off voltage / current		5 VDC / 2 mA	
Input impedance		About 2.4 kΩ	
Response time	Off → On	0.5 msec or less	
	On → Off	0.5 msec or less	
Common		1 points / 1 com	
Internal current consumption		65 mA	200 mA
Operation indicator		LED display	
External wiring		20 points terminal block connector (M3×6 screw)	38 points terminal block connector (M3×6 screw)
Weight		160 g	400 g
Wiring diagram			



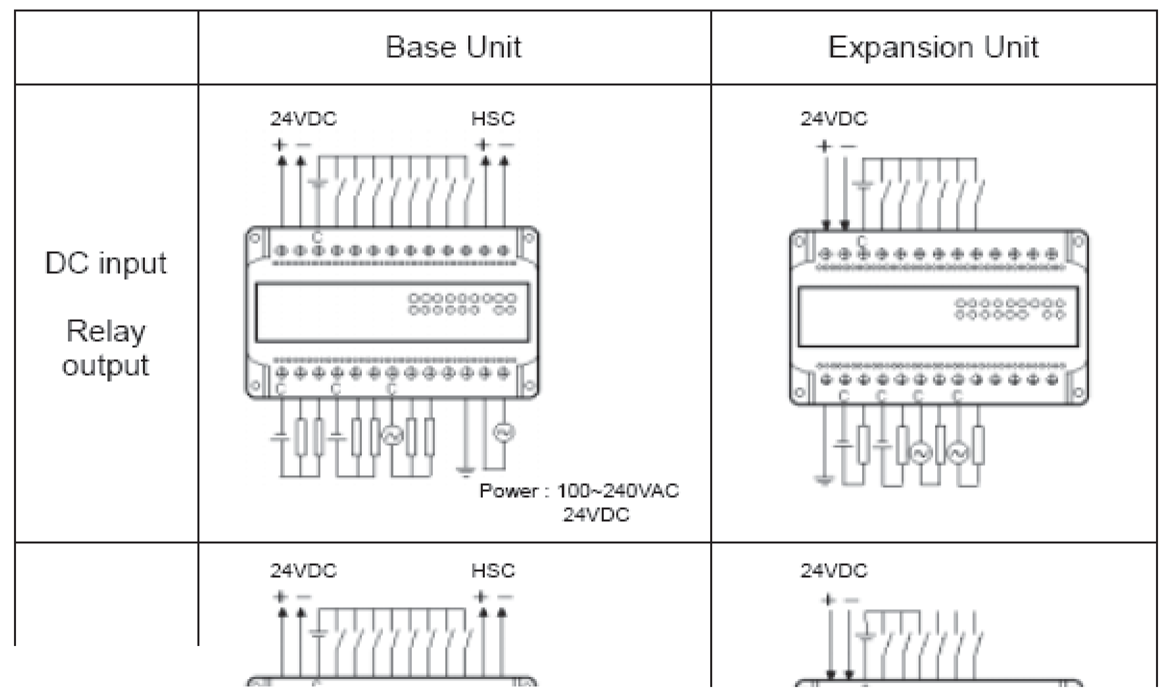
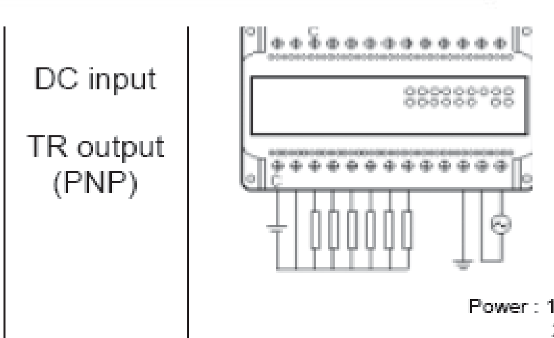
8 points relay output module

Item \ Type		K200S	
		K3Y-101S	
Output points		8 points	
Insulation method		Photo coupler	
Rated load voltage / current		24 VDC / 2A (resistive load), 220 VAC / 2A (cosψ = 1)	
Minimum load voltage / current		5 VDC / 1mA	
Maximum load voltage		125 VDC / 250 VAC	
Leakage current		0.1 mA (220 VAC, 60Hz)	
Maximum switching frequency		3,600 times / hour	
Surge absorber		None	
Lifetime of contact	Mechanical	No load	Over 20 million times
	Electrical	Rated voltage / current	Over 0.1 million times
		200VAC / 1.5A, 240VAC / 1A (cosψ = 0.7)	Over 0.1 million times
		200VAC / 1A, 240VAC / 0.5A (cosψ = 0.35)	Over 0.1 million times
		24VDC / 1A, 100VDC / 0.1A (L / R = 7ms)	Over 0.1 million times
Response time	Off → On	10msec or less	
	On → Off	12msec or less	
Common method		1 point / 1COM (Independent common)	
Internal current consumption		210mA (when all outputs are on)	
Operation indicator		LED	
External wiring		18 points terminal block connector (M3×6 screw)	
Weight		160 g	
Wiring diagram			



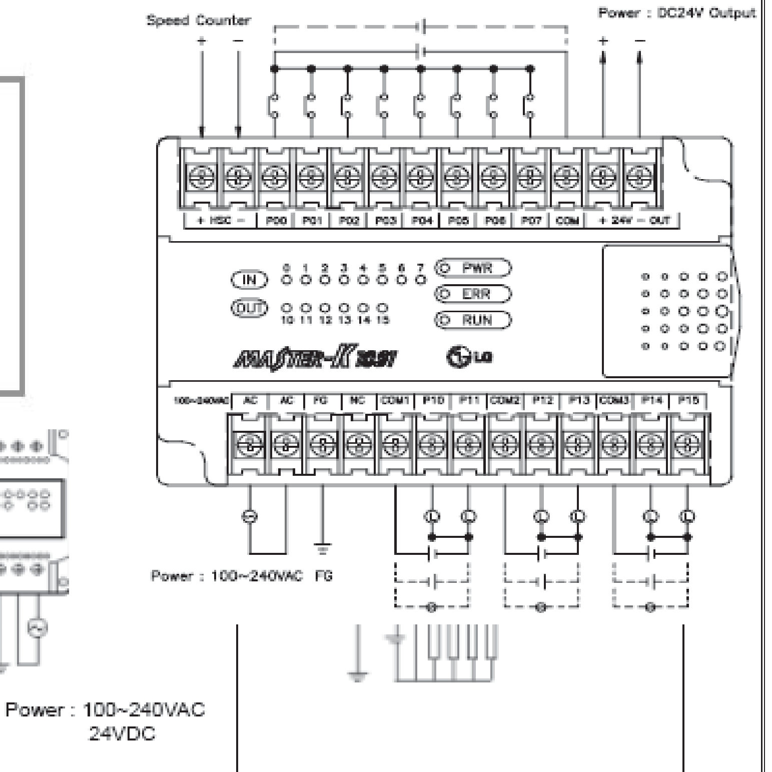
■ Wiring

- **K10S**

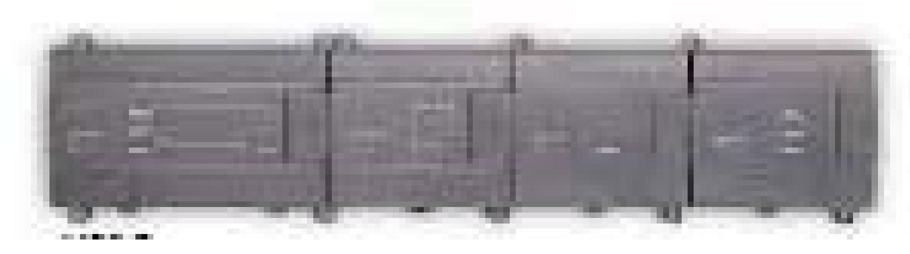
	Base Unit	Expansion Unit
DC input Relay output		
DC input TR output (PNP)	24VDC HSC 	24VDC

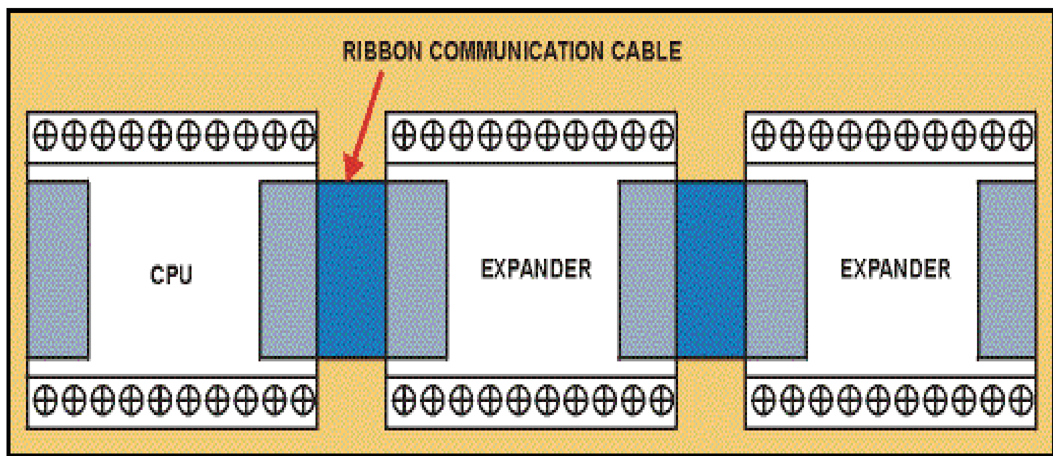
MASTER-K10S1



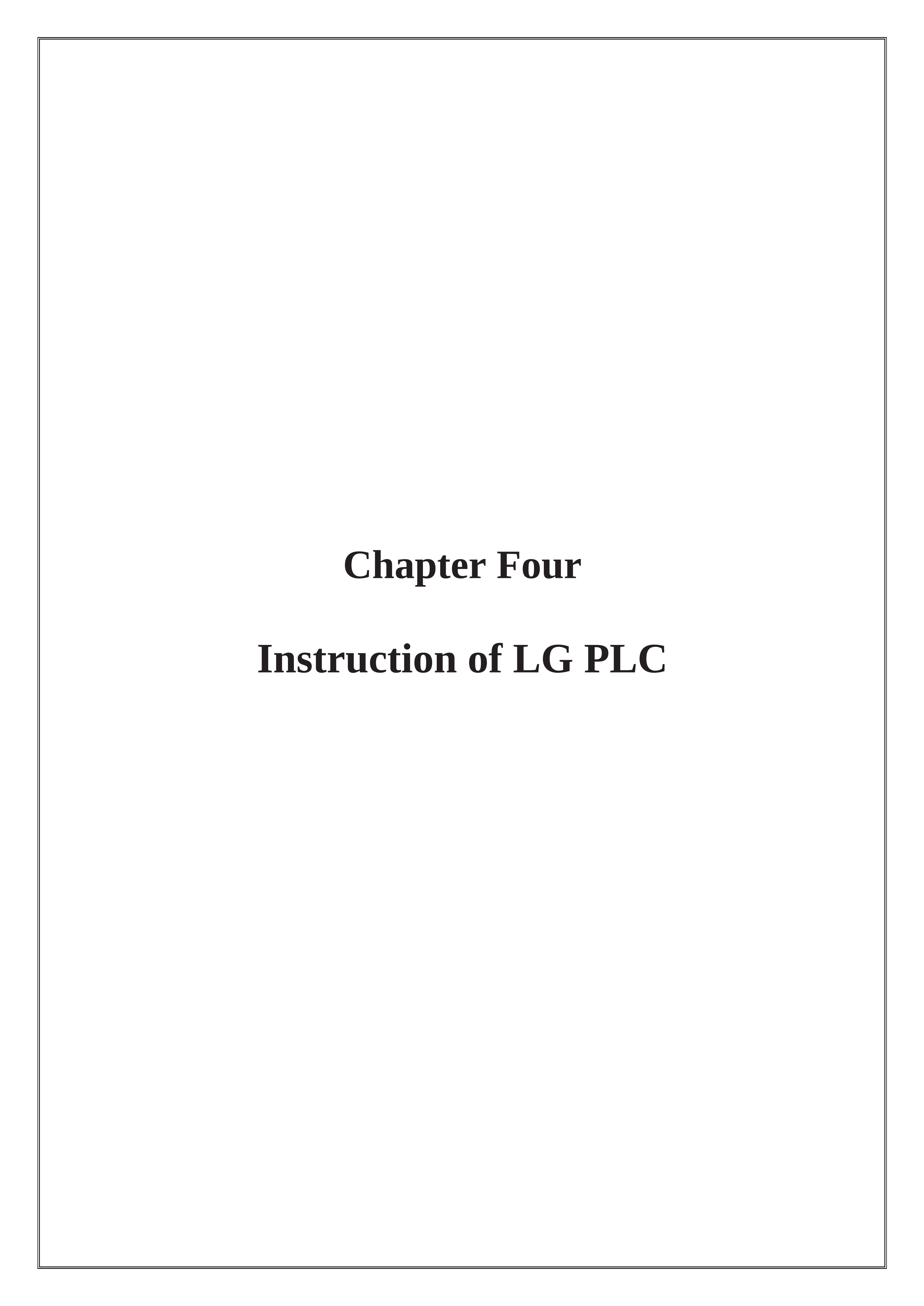


In some cases we get problem with the number of input & output and we solve this problem by expander or modular .









Chapter Four

Instruction of LG PLC

Contact instructions

Mnemonic symbol	Function No.	Ladder symbol	Unit	Contents of processing
LOAD	-		-	NO contact operation start
LOAD NOT	-		-	NC contact operation start
AND	-		-	NO contact series connection
AND NOT	-		-	NC contact series connection
OR	-		-	NO contact parallel connection
OR NOT	-		-	NC contact parallel connection

Connection instructions

Mnemonic symbol	Function No.	Ladder symbol	Unit	Contents of processing
AND LOAD	-	A B	-	Series connection of blocks
OR LOAD	-	A B	-	Parallel connection of blocks
MPUSH	005	MPUSH	-	Stores the operation result
MLOAD	006	MLOAD	-	Reads the operation result from MPUSH
MPOP	007	MPOP	-	Reads the operation result from MPUSH and clears the result

Inversion instruction

Mnemonic symbol	Function No.	Ladder symbol	Unit	Contents of processing
NOT	-	——✳——	-	Invert the operation result

Master control instructions

Mnemonic symbol	Function No.	Ladder symbol	Unit	Contents of processing
MCS	010	—[MCS n]—	-	Start a master control
MCSCLR	011	—[MCSCLR n]—	-	End a master control

Output instructions

Mnemonic symbol	Function No.	Ladder symbol	Unit	Contents of processing
D	017	—[D Ⓓ]—	-	Generates one scan pulse on the rising edge of input signal.
D NOT	018	—[D NOT Ⓓ]—	-	Generates one scan pulse on the falling edge of input signal.
SET	-	—[SET Ⓓ]—	-	Set a device
RST	-	—[RST Ⓓ]—	-	Reset a device
OUT	-	—( )—|	-	Output a device

Step controller instructions

Mnemonic symbol	Function No.	Ladder symbol	Unit	Contents of processing
SET S	-	[SET Sxx.xx]	-	Sequential processing control
OUT S	-	(Sxx.xx)	-	Last-in priority control

END instruction

Mnemonic symbol	Function No.	Ladder symbol	Unit	Contents of processing	CPU	Page
END	001	[END]	-	Ends a sequence program	○	4– 25

No operation instruction

Mnemonic symbol	Function No.	Ladder symbol	Unit	Contents of processing
NOP	000	No ladder symbol	-	No operation (occupies 1 step)

Bit contact instructions

Mnemonic symbol	Function No.	Ladder symbol	Unit	Contents of processing
BLD	248	B D n	-	NO contact operation start with the n^{th} bit of [D]
BLDN	249	BN D n		NC contact operation start with the n^{th} bit of [D]
BAND	250	B D n		NO contact series connection with the n^{th} bit of [D]
BANDN	251	BN D n		NC contact series connection with the n^{th} bit of [D]
BOR	252	B D n		NO contact parallel connection with the n^{th} bit of [D]
BORN	253	BN D n		NC contact parallel connection with the n^{th} bit of [D]
BOUT	236	BOUT D n		Output the result of operation to the n^{th} bit of [D]
BSET	223	SET D n		Set the n^{th} bit of [D]
BRST	224	RST D n		Clear the n^{th} bit of [D]

The kinds of PLC Commands

Applicable Commands in LG PLC K Series :

Sequence Command :
Basic Commands for creating Sequence Logic Circuits.

Comparison Command :
Application Commands to execute the Comparison Operations.

Arithmetic Command :
Application Commands to execute the Arithmetic Operations.

Logical Operation Command :
Application Commands to execute the Logical Operations.

Rotate/Shift Command :
Application Commands to rotate or shift Data.

Increment/Decrement :
Application Commands to add or subtract ‘ ١ ’ to the data.

Conversion Command :
Application Commands to change the type of Data.

Transfer Command :
Application Commands to copy, exchange or transfer Data between the internal devices.

Timer/Counter Command :
Basic Commands to use Timer and Counter.

Jump/Interrupt Command :
Application Commands to execute Interrupt, Call or Jump with the specified program.

Sequence Command

- **NOP** No Operation ١ Step

Function :

١. This is a No Operation Command and it doesn’t effect to the operation results of the program.
٢. This command shows only in the Mnemonic program.

Objects for NOP use :

١. For occupying the space of the command to be used later.
٢. For removing Command with keeping the number of steps temporarily.

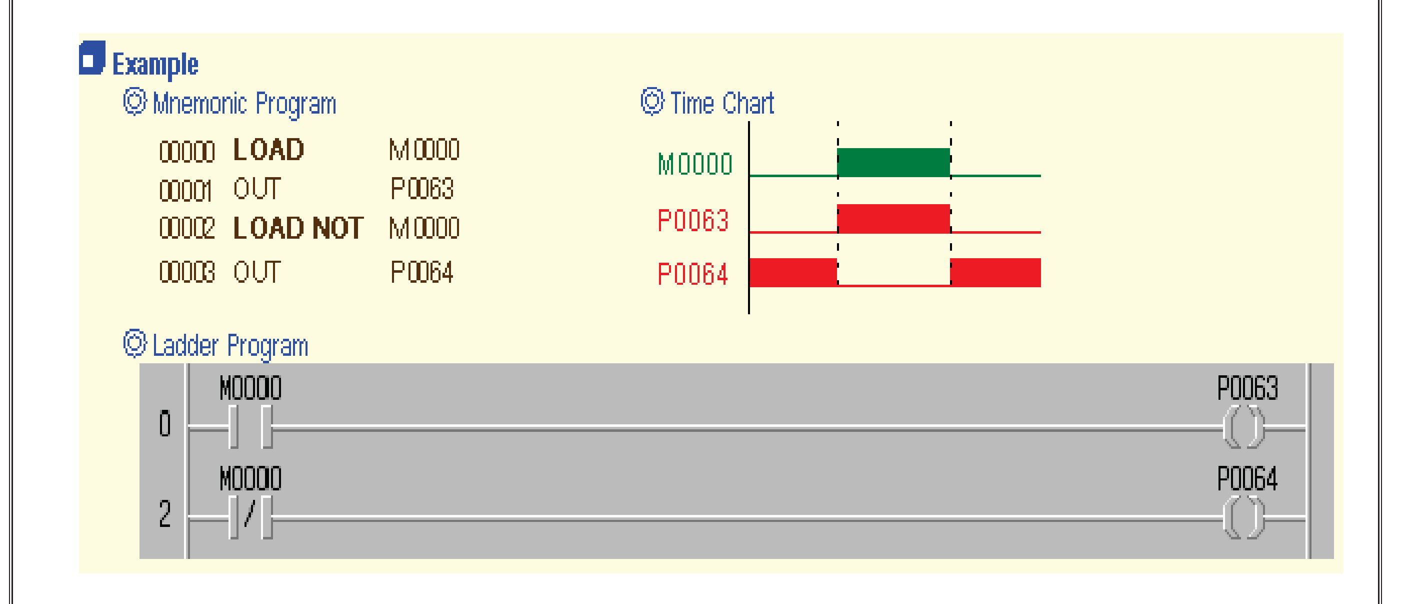
Note :
‘Optimize Program’ Function in Edit Menu of KGL for Windows is used for deleting all NOP Commands in the program automatically.

- **LOAD** NO contact ١ Step
- **LOAD NOT** NC Contact ١ Step

Function :

LOAD : Starts NO (Normally Open) contact.
Executes the ON/OFF operation of the specified device.

LOAD NOT : Starts NC (Normally Closed) contact.
Executes the converted ON/OFF operation of the specified device.



- **AND** — **Serial Connection of NO Contact** — **١ Step**
- **AND NOT** — **Serial Connection of NC Contact** — **١ Step**

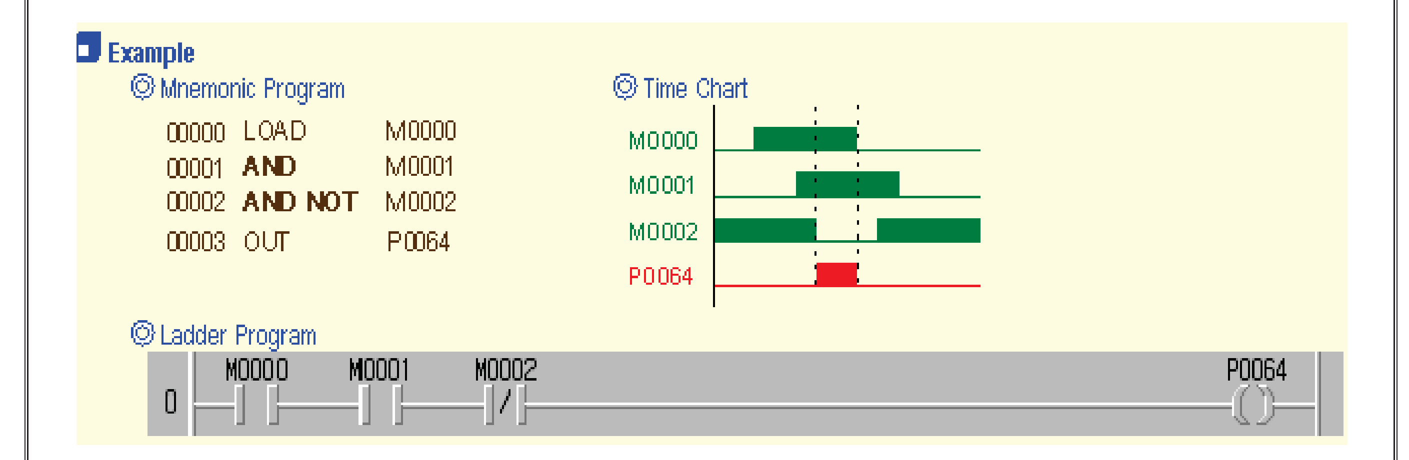
Function :

AND : **Serial connection of Normally Open Contact.**

Performs a logic AND between the operand and the Boolean result of The preceding command.

AND NOT : **Serial connection of Normally Closed Contact.**

Performs a logic AND NOT between the operand and the Boolean result of the preceding command.



- **OR** **Parallel Connection of NO Contact** **١ Step**
- **OR NOT** **Parallel Connection of NC Contact** **١ Step**

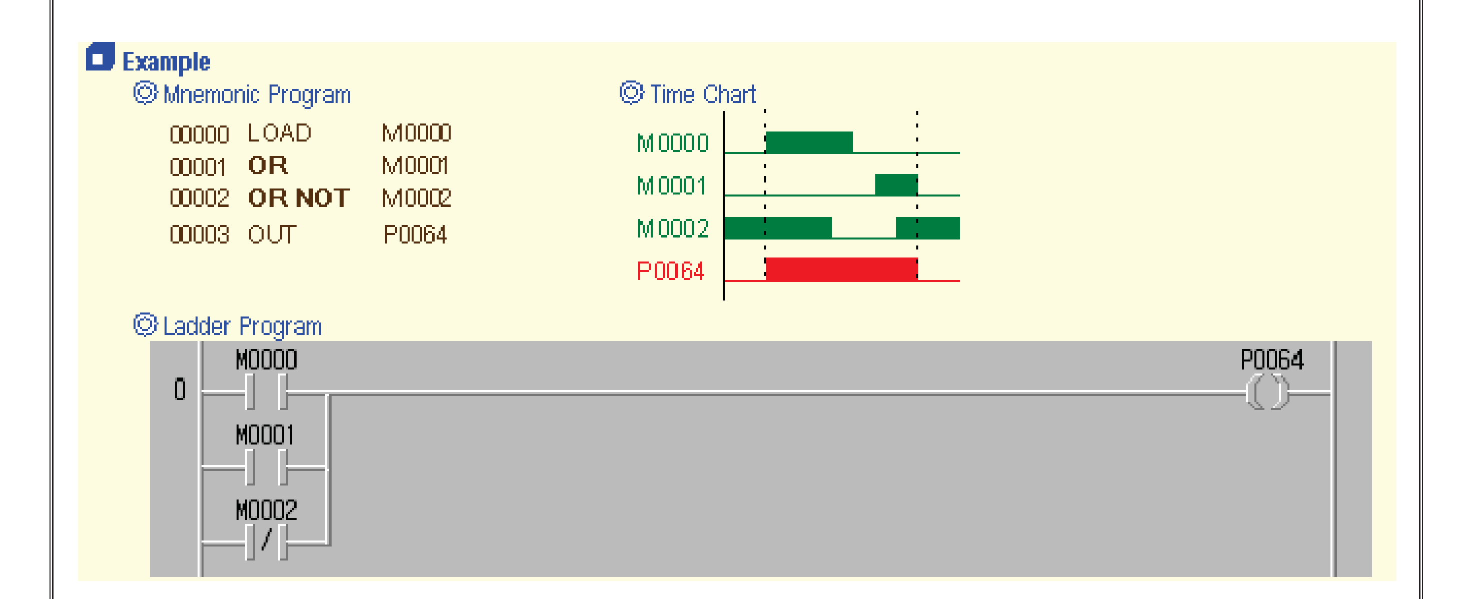
Function :

OR : **Parallel connection of NO contact.**

Performs a logic OR between the operand and the Boolean result of the preceding command.

OR NOT : **Parallel connection of NC contact.**

Performs a logic OR NOT between the operand and the Boolean result of the preceding command.

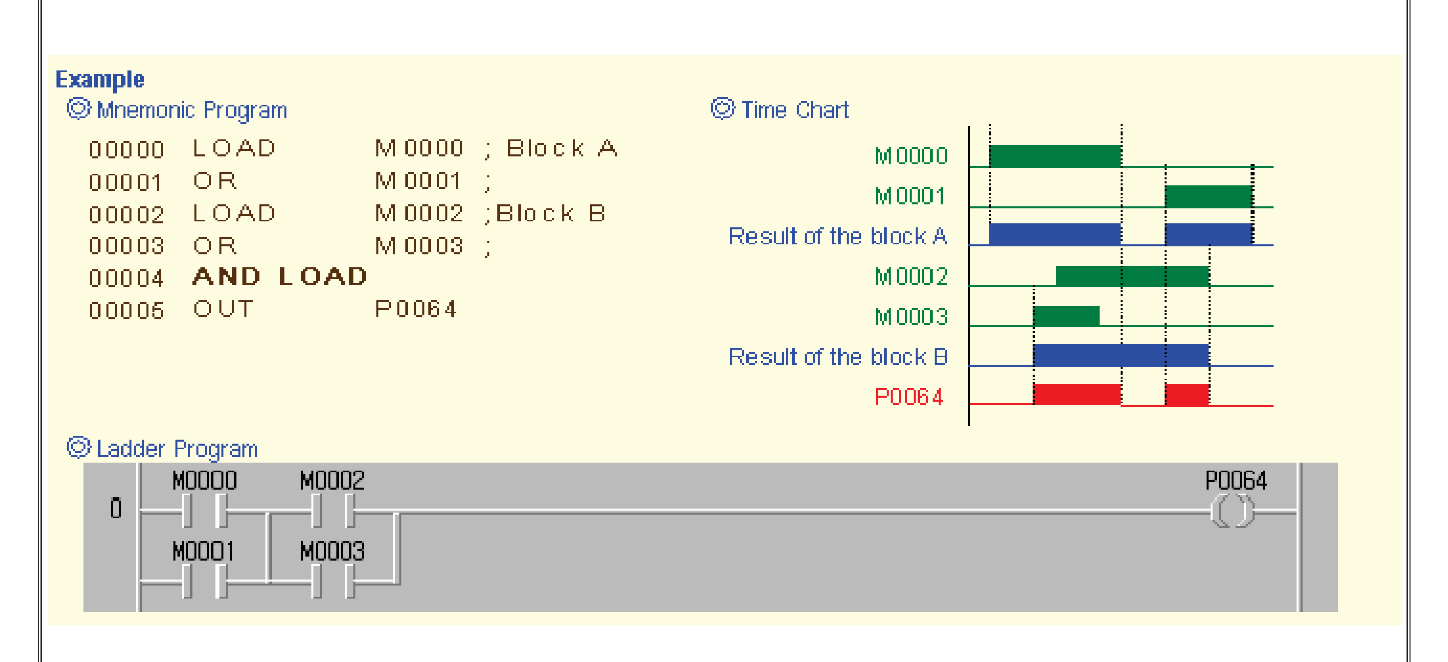


- **AND LOAD** **Serial Block Operation** **١ Step**

Function :

١. AND Operation of the result of A Block and B Block.

٢. It does not have its own ladder expression and it is converted to the mnemonic program in the necessary circuit.

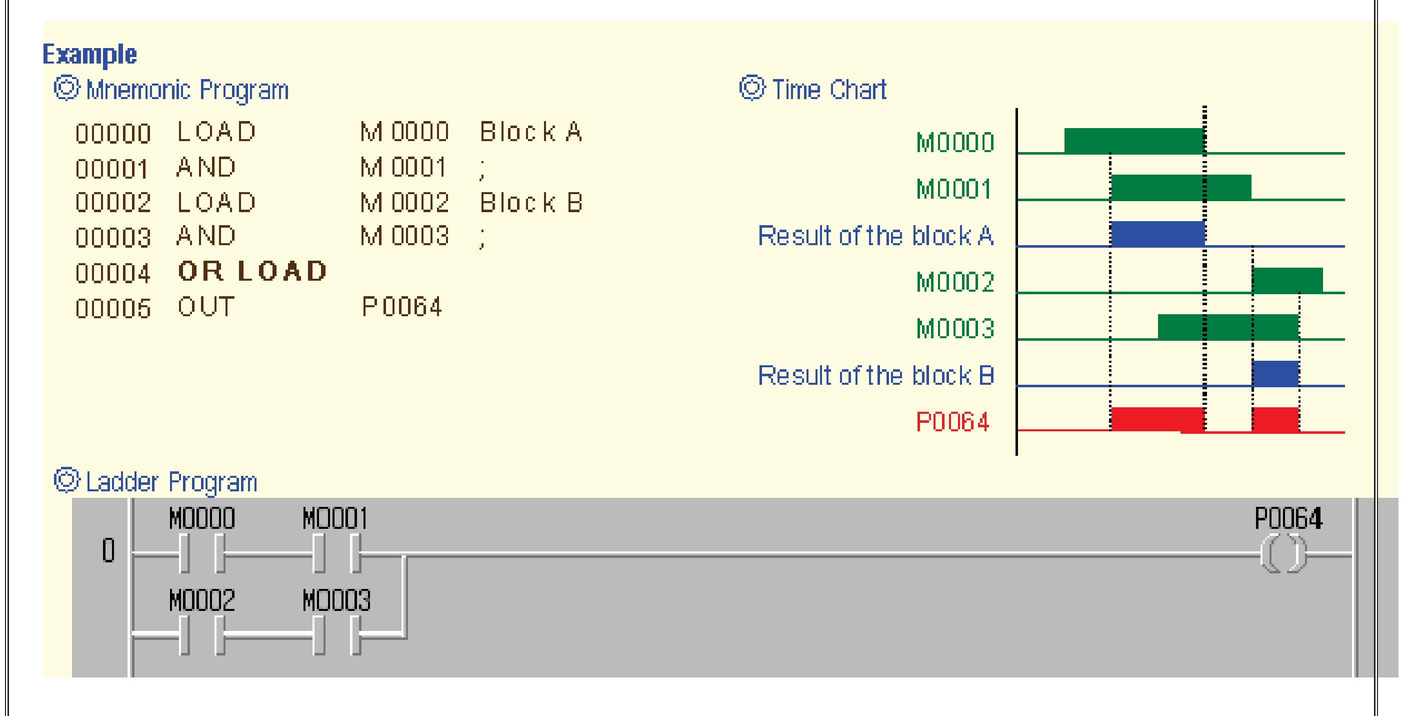


OR LOAD **Parallel Block Operation** **١ Step**

Function :

١.OR Operation of the result of A Block and B Block.

٢. It does not have its own ladder expression and it is converted to the mnemonic program in the necessary circuit.



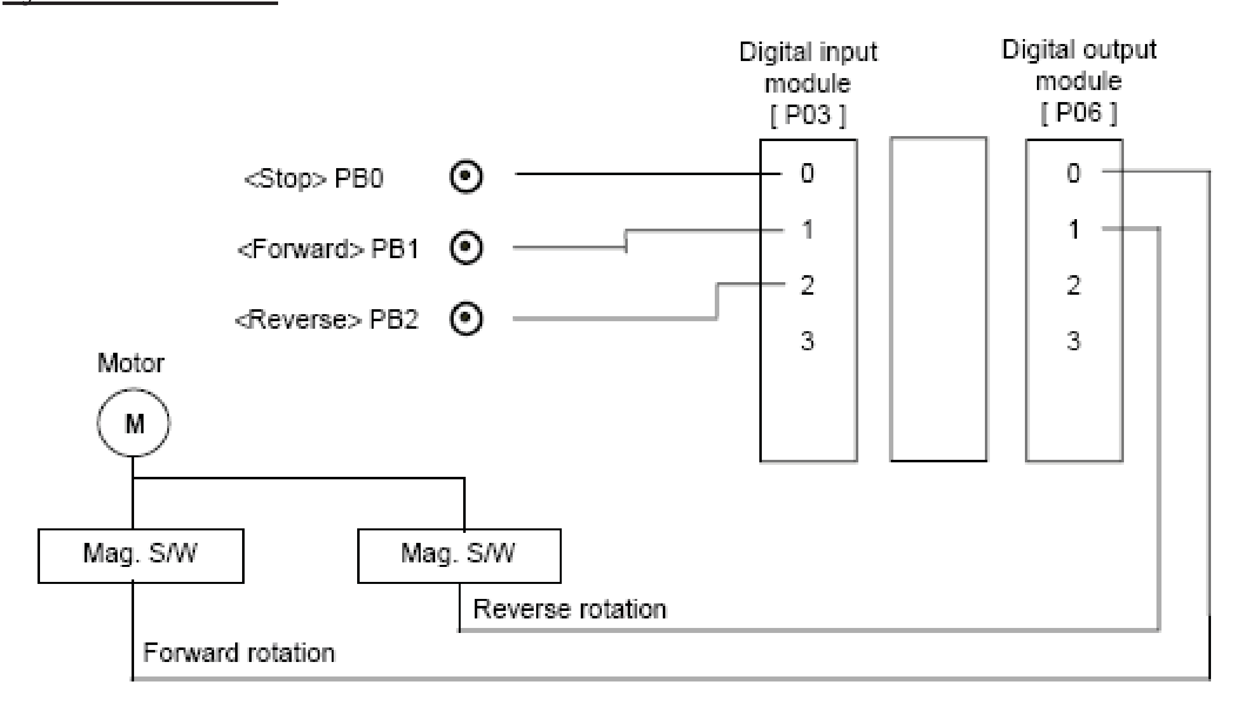
Example no. ١

The motor operation (Example of LOAD, AND, OR, OUT instructions)

Operation :

There are three push-button switches - PB٠, PB١, and PB٢. When PB١ is pushed, a motor will start to rotate with a forward (clockwise) direction. It will start to rotate with a reverse (counterclockwise) direction when the PB٢ is pushed. The PB٠ is emergency stop switch and the motor will stop operation when the PB٠ is pushed.

System structure :



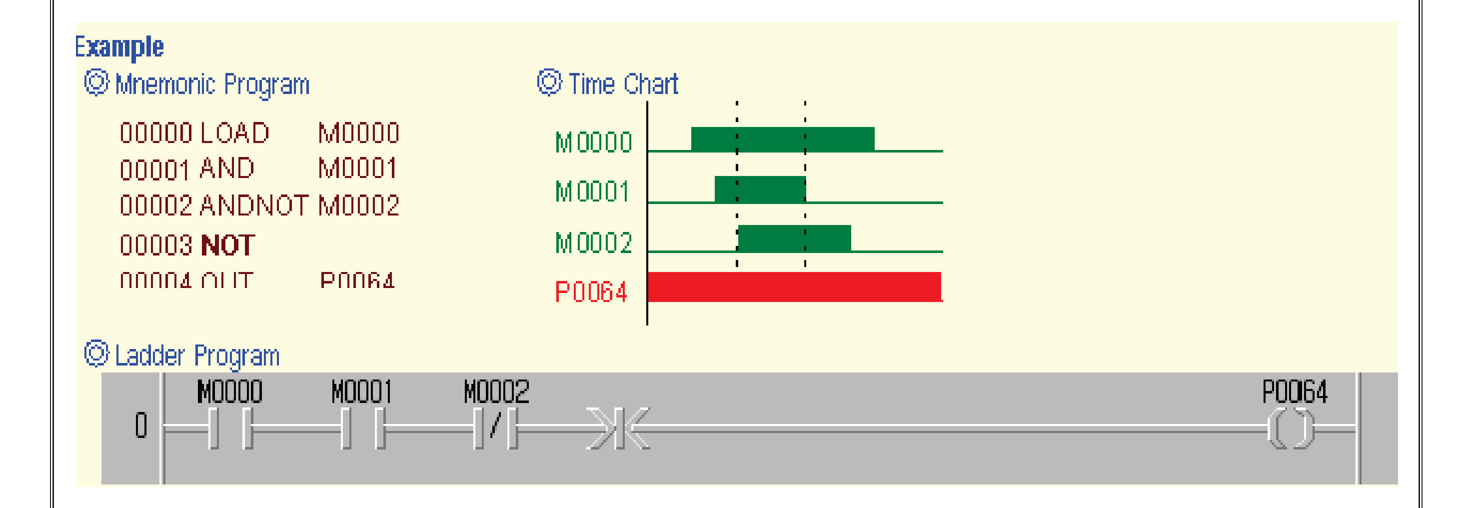
Program :

```
0000 | P031                P030   P032   P061
     |--| |--+------------|/|----|/|----|/|-------( P060 )--|
     | P060  |
     |--| |--+
0006 | P032                P030   P031   P060
     |--| |--+------------|/|----|/|----|/|-------( P061 )--|
     | P061  |
     |--| |--+
0011 |---------------------------------------------[ END ]--|
```

NOT Conversion Command ١ Step

Function :

This command negates the Boolean result of the preceding command.



Example no. ٢

Program example

The following two programs perform same operation.

Program A

P020 P021 P022 P023 P024 (P060)

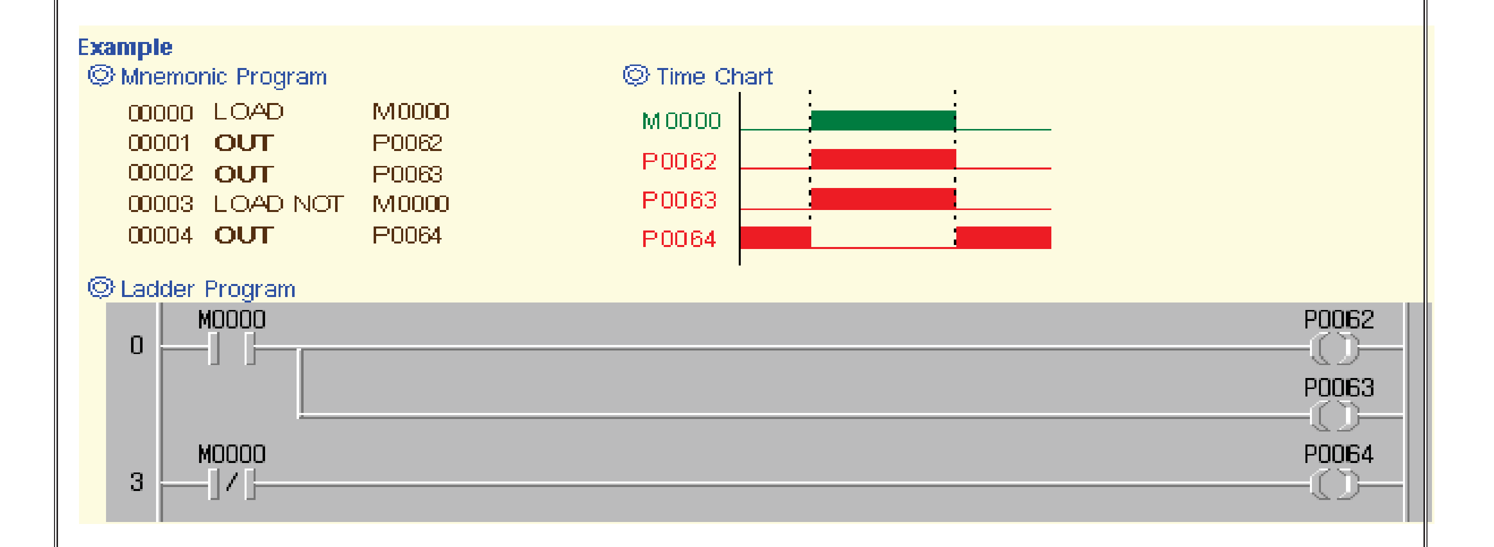
Program B

P020 P021 P022 P023 P024 (P060)

- **OUT Output the Result ١ Step**

Function :

The operation result up to 'OUT' command is outputted to the specified device. Parallel circuit is available in OUT Command.

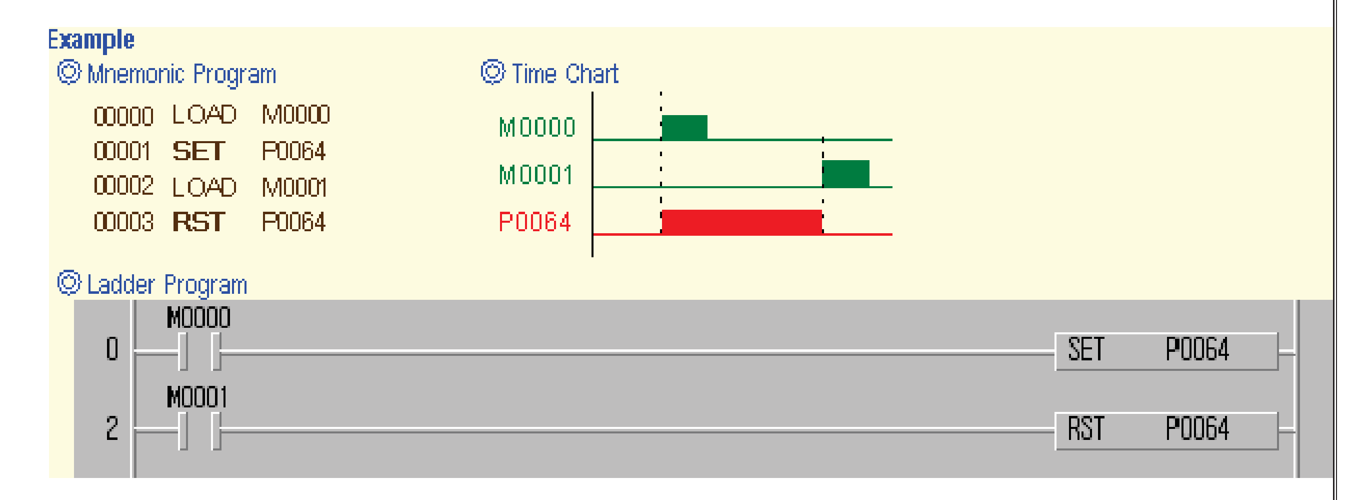


- **SET Self holding with ON status ١ Step**
- **RST Self holding with OFF status ١ Step**

Function :

SET : When the input condition gets ON, keeps the specified device d as ON status and keeps the device d as ON status even if the input condition gets OFF.

RST : When the input condition gets OFF, keeps the specified device d as ON status and keeps the device d as OFF status even if the input condition gets OFF.



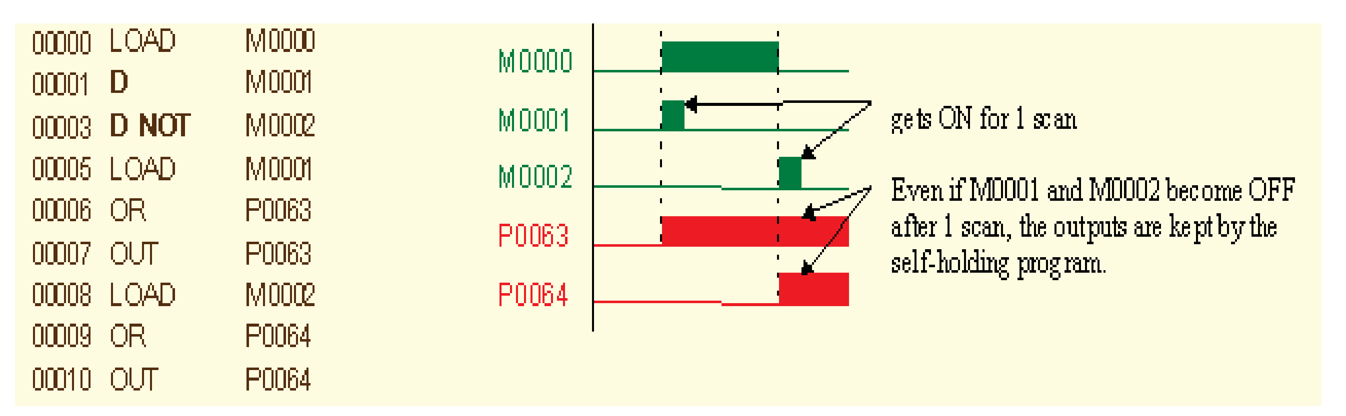
- **D** **Derivative Command** **٢ Steps**
- **D NOT** **Derivative Command (Conversion)** **٢ Steps**

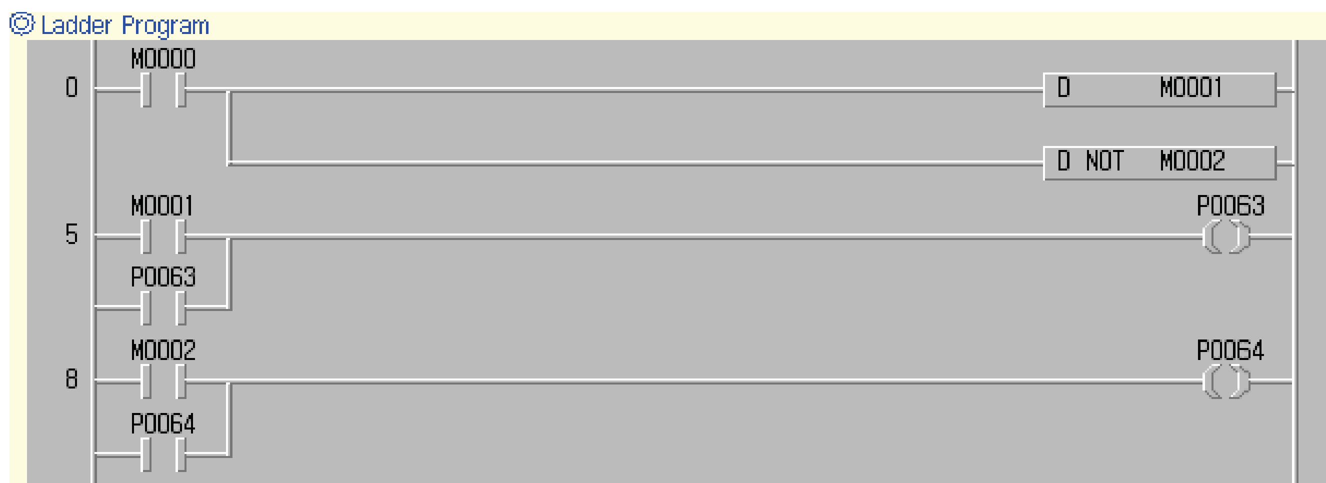
Function :

This is a Derivative command for detecting the point the input condition is changed. When the change of the input condition is detected, the specified device(d) becomes ON for ١ scan.

D : When the input condition changes from OFF to ON, the specified device becomes ON for ١ scan.

D NOT : When the input condition changes from ON to OFF, the specified device becomes ON for ١ scan.





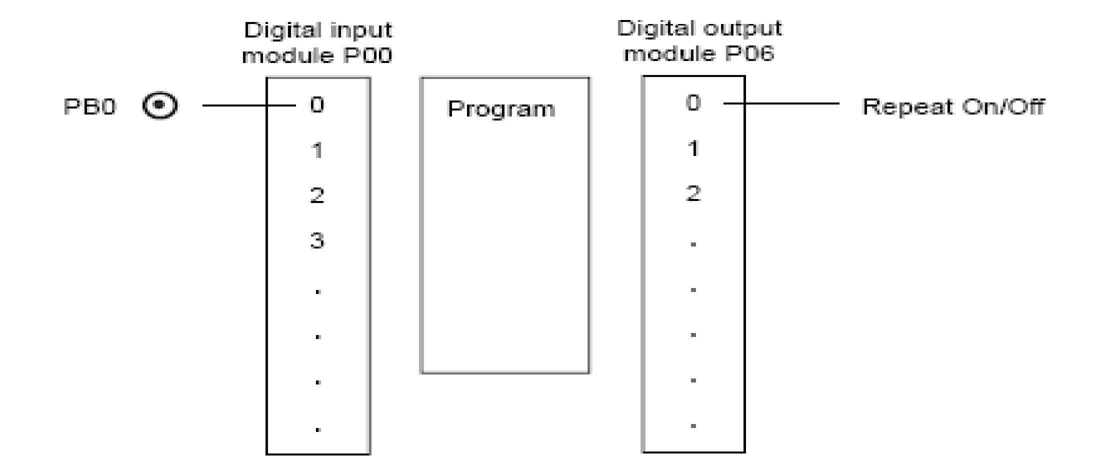
Example no. ٣

The on / off toggle control (Example of D instructions)

Operation :

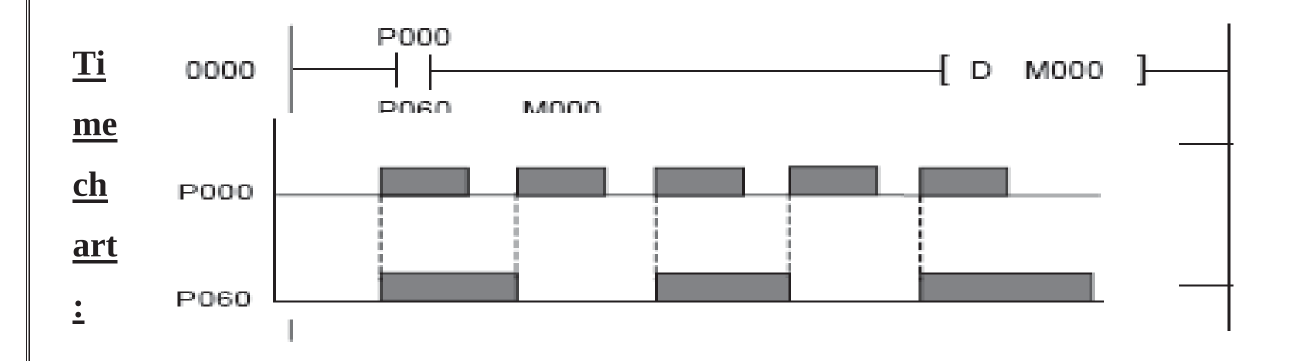
When the push-button PB٠ is pushed, the P٠٦٠ is switched on. It is switched off when the PB٠ is pushed again. The P٠٦٠ will repeat on / off whenever the PB٠ is pushed.

System structure :



Program :

Time chart :

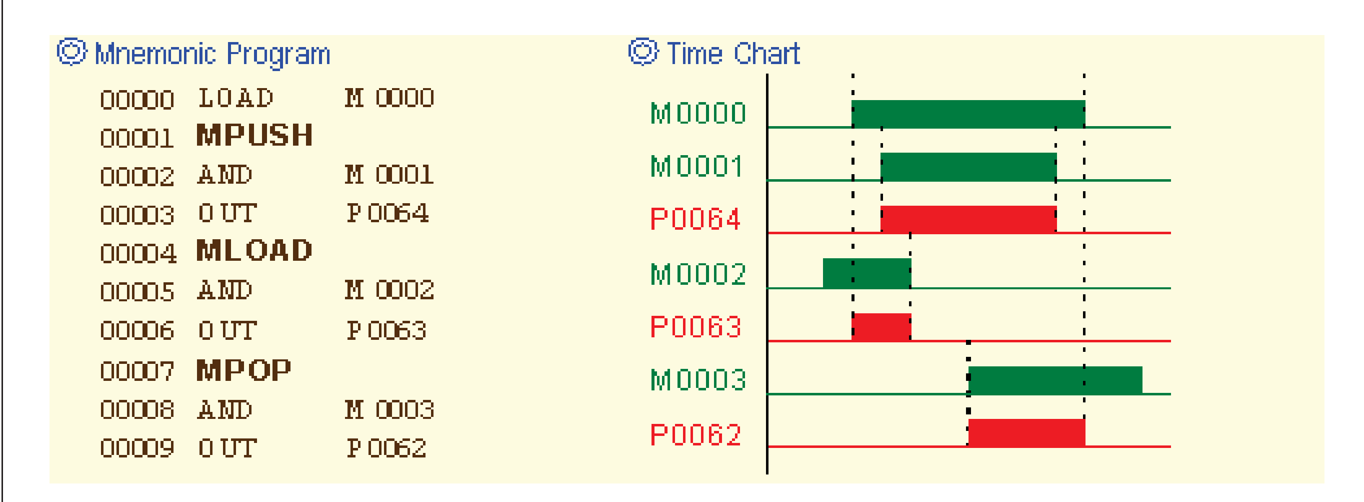


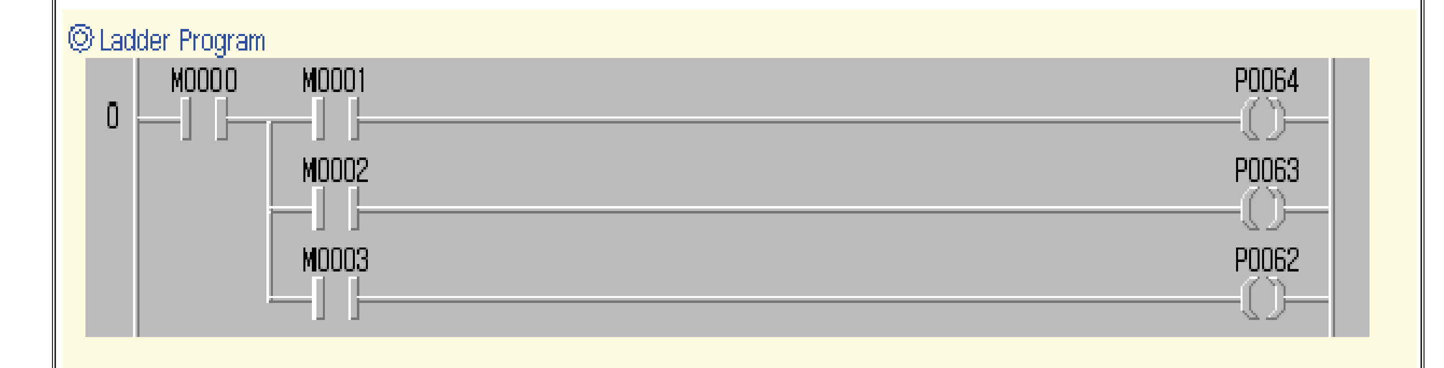
- **MPUSH** **Push the Result** **١ Step**
- **MLOAD** **Load the Pushed Result** **١ Step**
- **MPOP** **Pop the Result** **١ Step**

Function :

Multiple branches in the ladder diagram can be drawn by this command.

MPUSH shall be used with MPOP all the time.





- **MCS** **Master Control** **١ Step**
- **MCSCLR** **Master Control Clear** **١ Step**

Function :

١. When the input condition of MCS gets ON, it executes until MCSCLR which is identical to MCS number (n).

٢. When the input condition of MCS gets OFF, it doesn't executes MCSCLR which is identical to MCS number (n).

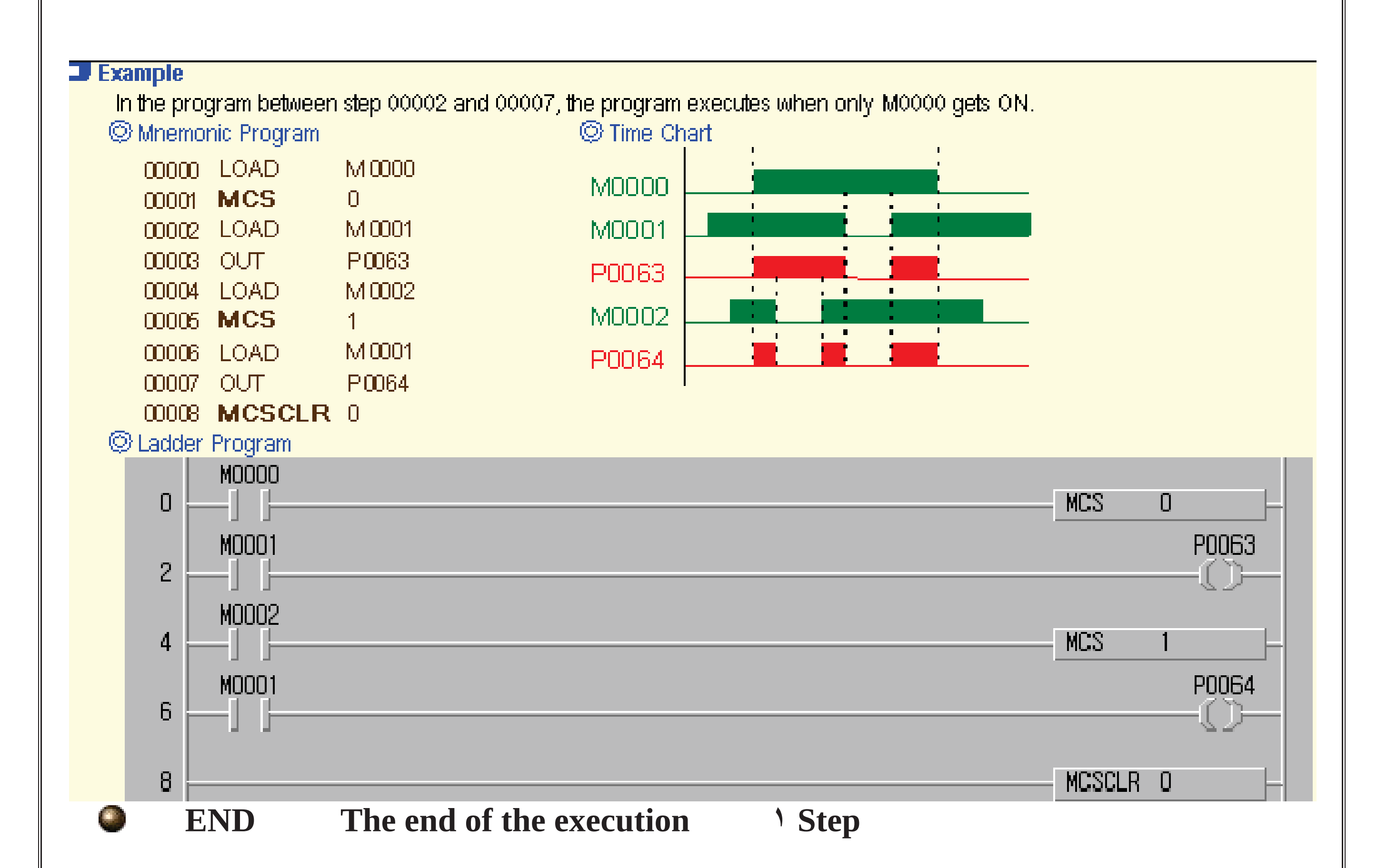
٣. The priority of MCS is from ٠(highest priority) to ٧(lowest priority) in order.

٤. MCS with the higher priority is used first and MCSCLR with the lower priority is used first.

٥. If a user clears the master control with the higher priority (MCSCLR), the ones with the lower priority are cleared.

٦. If MCS number is doubled before MCSCLR is executed, It causes an error.

٧. If the priority order of MCS or MCSCLR is changed, it also causes an error.



END **The end of the execution** ١ **Step**

Function :

١. Indicates the end of the program.

٢. The scan of the program starts from step ٠٠٠٠ again after the execution of the END Command.

٣. END Command must be at the end of the program. If not, an error is occurred.

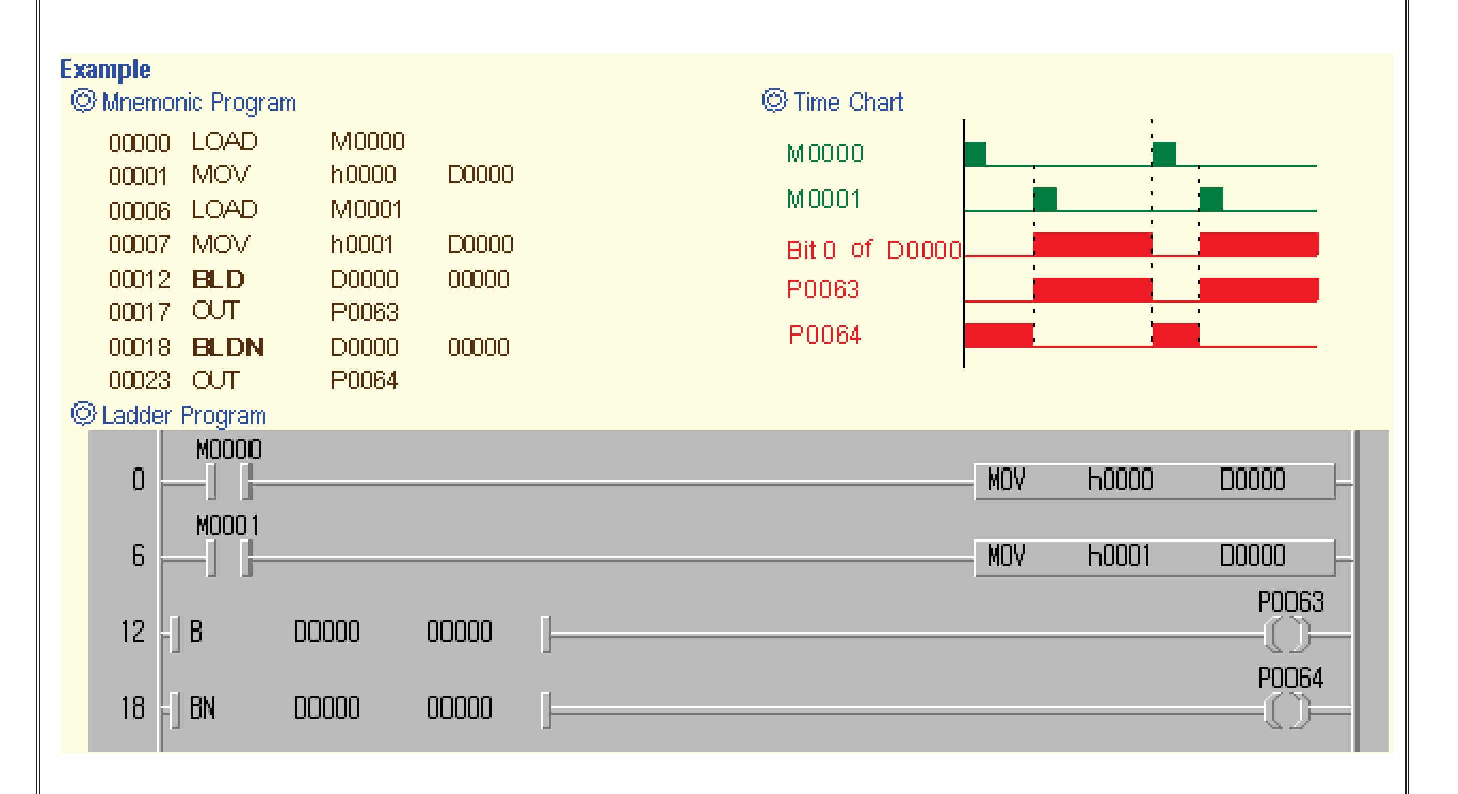


- **BLD** d n **Bit LOAD** ٥ **Steps**
- **BLDN** d n **Bit LOAD NOT** ٥ **Steps**

Function :

BLD : Load the nth bit of the specified area d as the result of the current operation.

BLDN : Load the reversed nth bit of the specified area d as the result of the current operation.



- **BAND** d n **Bit AND** ٥ **Steps**
- **BANDN** d n **Bit AND NOT** ٥ **Steps**

Function :

BAND : AND the nth bit of the specified area d as the result of the current operation.



BANDN : AND the reversed nth bit of the specified area d as the result of the current operation.

Mnemonic Program

```
00000  LOAD   M0000
00001  MOV    h0000   D0000
00006  LOAD   M0001
00007  MOV    h0001   D0000
00012  LOAD   M0002
00013  BAND   D0000   00000
00018  OUT    P0063
00019  LOAD   M0002
00020  BANDN  D0000   00000
00025  OUT    P0064
```

Time Chart

M0000
M0001
Bit 0 ofD0000
M0002
P0063
P0064

Ladder Program

```
0   M0000 —| |—————————————— MOV h0000 D0000
6   M0001 —| |—————————————— MOV h0001 D0000
12  M0002 —| |—| B  D0000 00000 |————— ( ) P0063
19  M0002 —| |—| BN D0000 00000 |————— ( ) P0064
```

- **BOR d n Bit OR ٥ Steps**
- **BORN d n Bit OR NOT ٥ Steps**

<u>Function :</u>

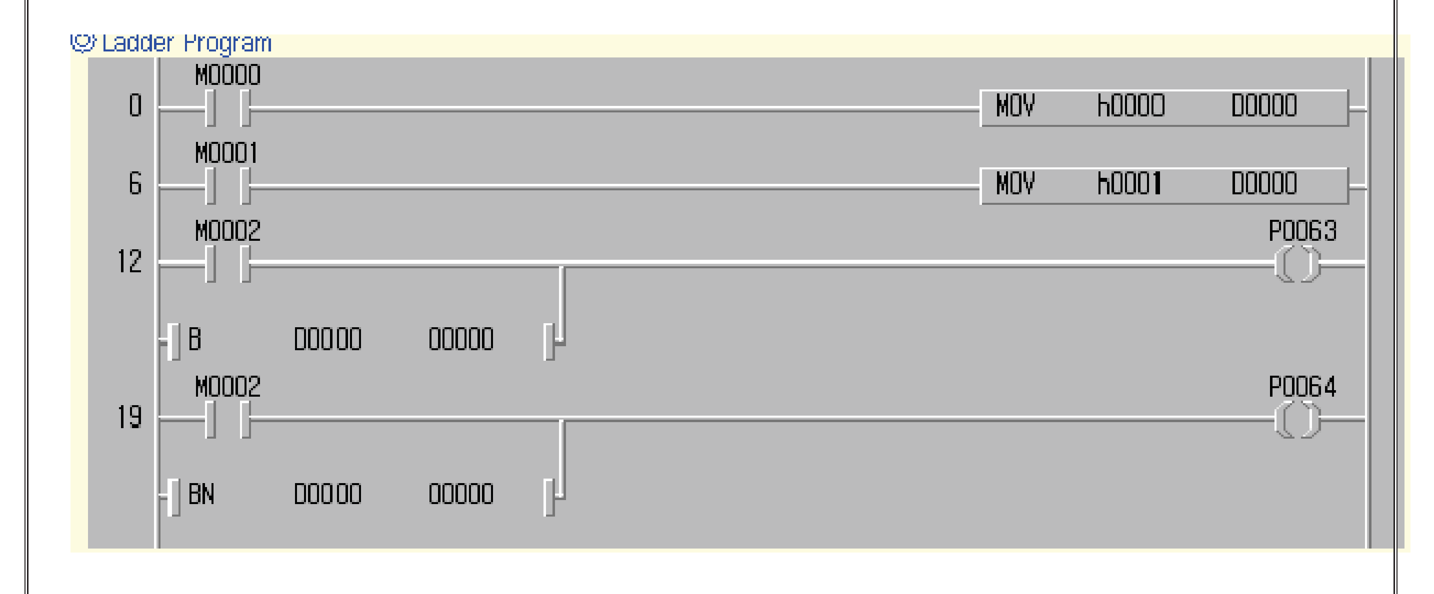
BOR :OR the nth bit of the specified area d as the result of the current operation.

BORN :OR the reversed nth bit of the specified area d as the result of the current operation.

```
00000  LOAD   M0000
00001  MOV    h0000   D0000
00006  LOAD   M0001
00007  MOV    h0001   D0000
00012  LOAD   M0002
00013  BOR    D0000   00000
00018  OUT    P0063
00019  LOAD   M0002
00020  BORN   D0000   00000
00025  OUT    P0064
```

M0000
M0001
Bit 0 of D0000
M0002
P0063
P0064

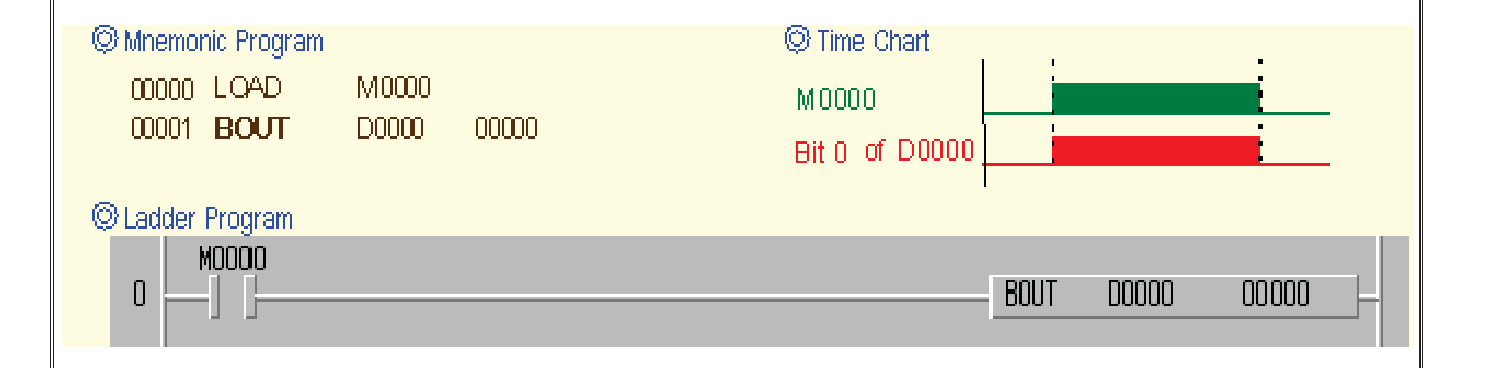




● **BOUT d n Bit OUT ٥ Steps**

Function :

Outputs the result of the current operation to the nth bit of the specified area.



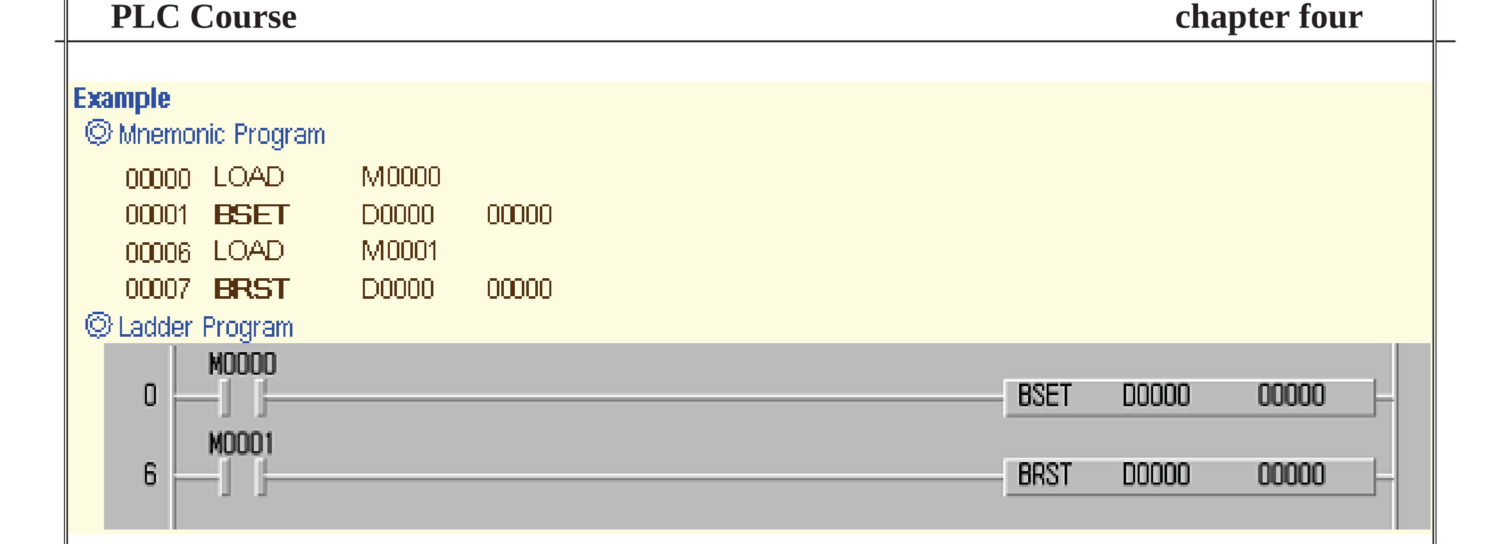
● **BSET d n Bit Set ٥ Steps**

● **BRST d n Bit Reset ٥ Steps**

Function :

BSET : **Set the nth bit of the specified area d.**

BRST : **Reset the nth bit of the specified area d.**



SET Sxx.xx Step Controller(Sequence Control) ٢ Steps

<u>The characteristics of the step controller :</u>

١. Self-holding Function – The current step is kept so long as there is no next command.

٢. Interlock – Only one step is outputted among ١٠٠ steps.

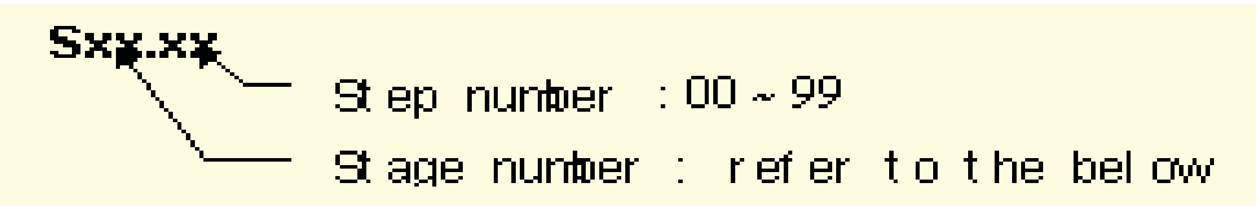
<u>Function :</u>

١. In a same stage, when the very previous step number gets ON, the current step number becomes ON as well

٢. Once the current step number is ON, it becomes self-holding status. and so, even if the input gets OFF, ON status is kept.

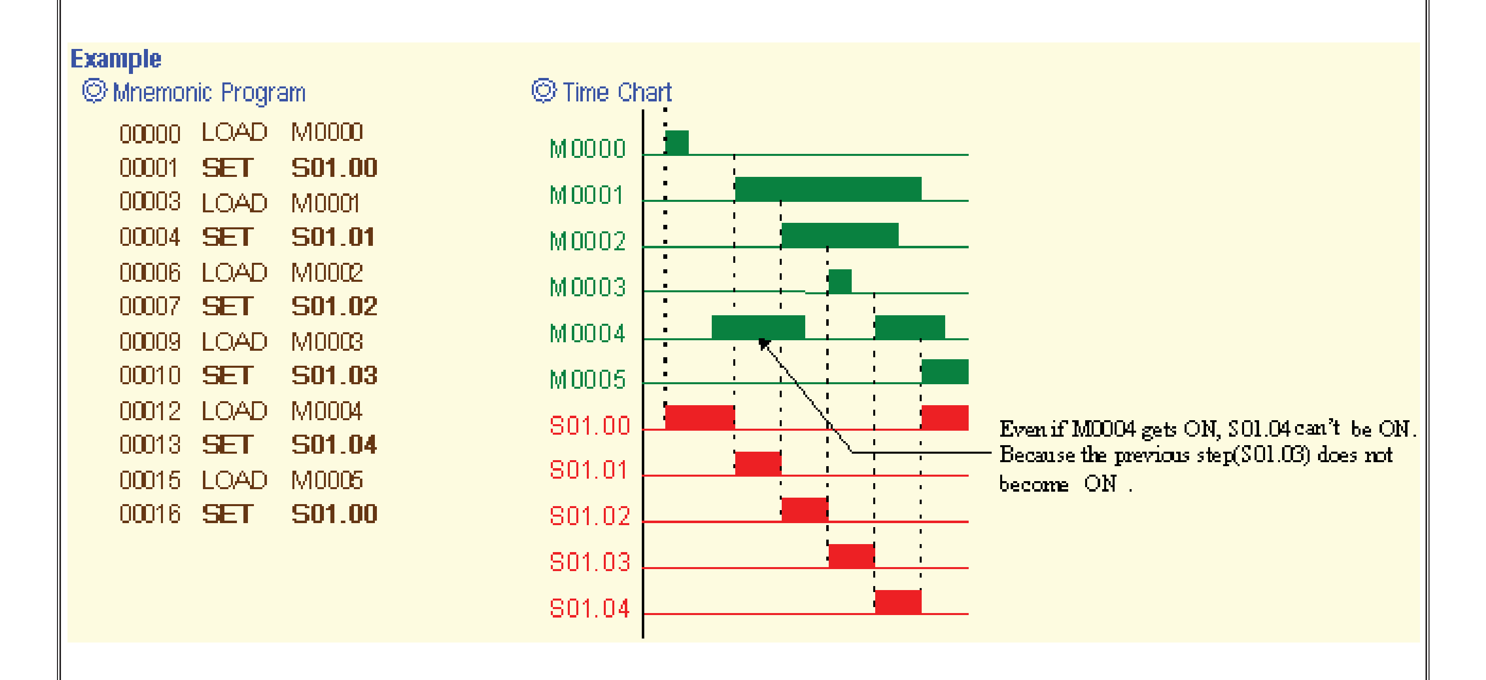
٣. Even if many input conditions are ON at the same time, only one step number gets ON in one stage.

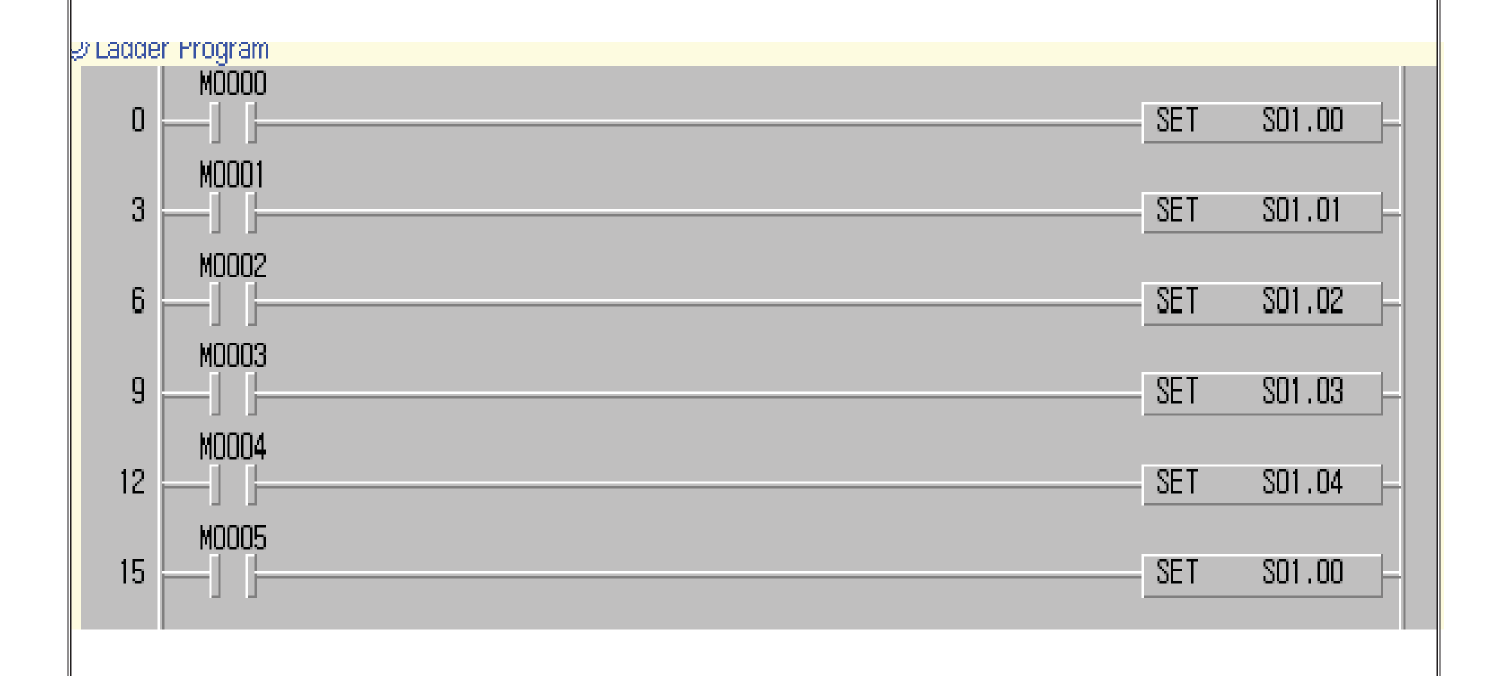
٤. To clear all steps(Sxx.xx) of the relevant stage, use ٠٠ step(Sxx.٠٠) of the stage.



Available ranges of the stage number :

PLC Type	Volatile Area	Nonvolatile Area
K١٠S١	S٠٠.xx ~ S١١.xx	S١٢.xx ~ S١٥.xx





OUT Sxx.xx Step Controller(Last-in Priority) ٢ Steps

The characteristics of the step controller :

١. Self-holding Function – The current step is kept so long as there is no next command.

٢. Interlock – Only one step is outputted among ١٠٠ steps.

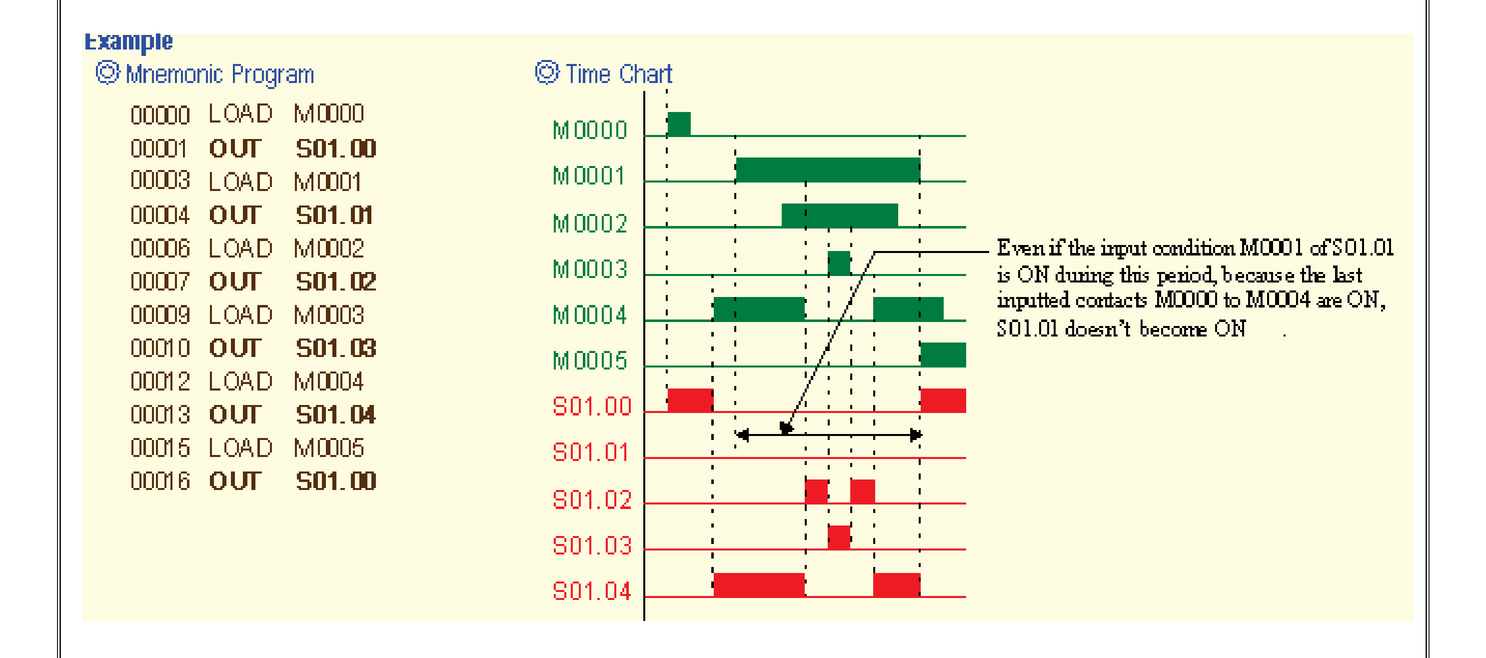
Function :

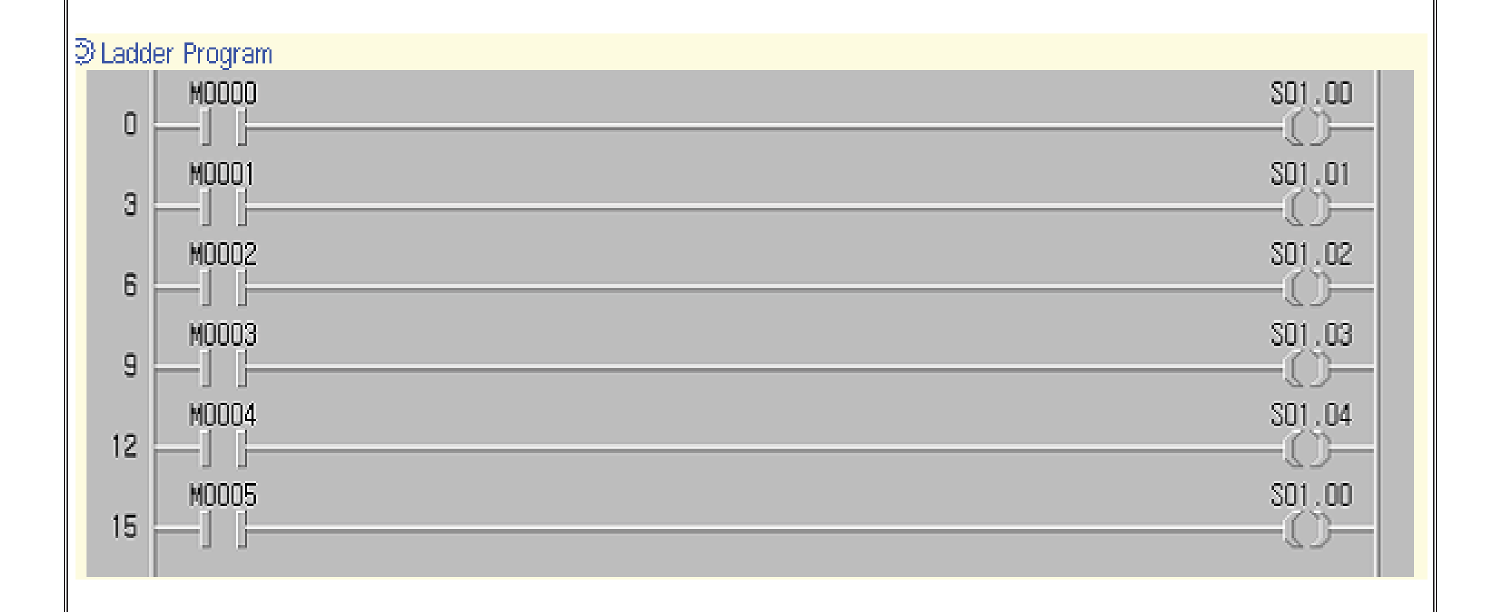
١. In a same stage, when the very previous step number gets ON, the current step number becomes ON as well

٢. Once the current step number is ON, it becomes self-holding status. and so, even if the input gets OFF, ON status is kept.

٣. Even if many input conditions are ON at the same time, only one step number gets ON in one stage.

٤. To clear all steps(Sxx.xx) of the relevant stage, use ٠٠ step(Sxx.٠٠) of the stage.

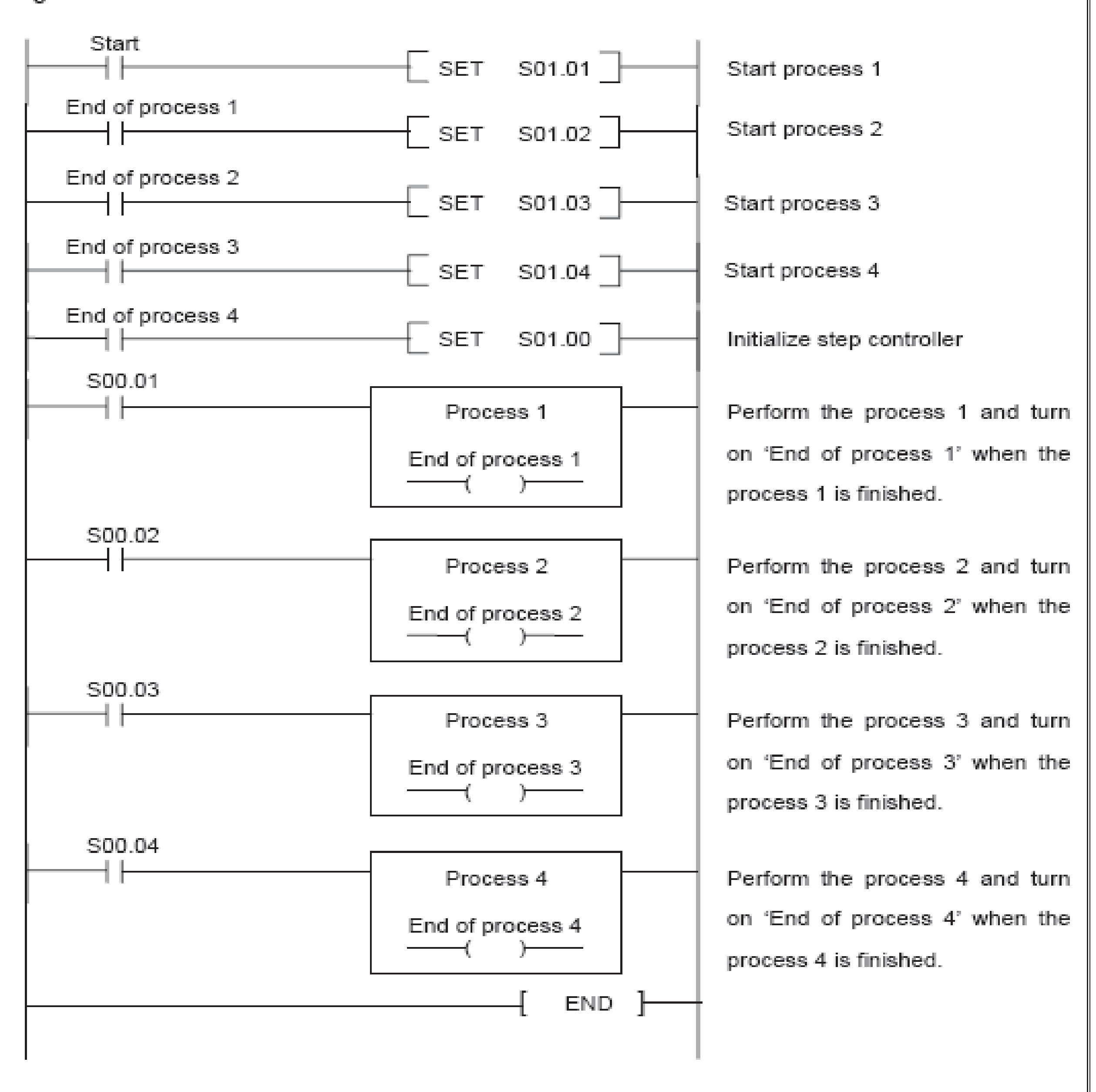



Ladder Program
0
M0000
S01.00
3
M0001
S01.01
6
M0002
S01.02
9
M0003
S01.03
12
M0004
S01.04
15
M0005
S01.00

The sequential control (example of SET Sxx.xx instruction) : -

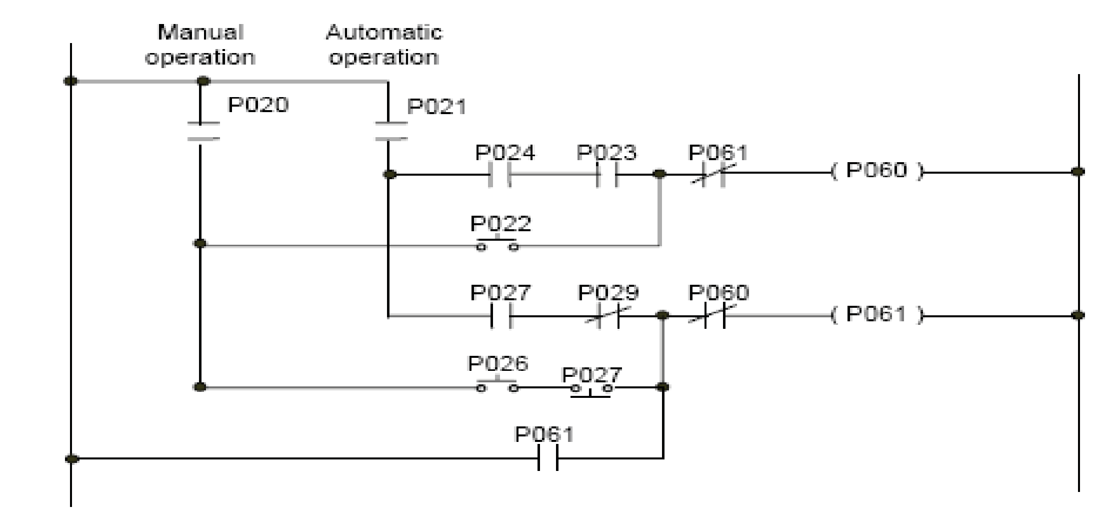
١. <u>Operation : -</u>

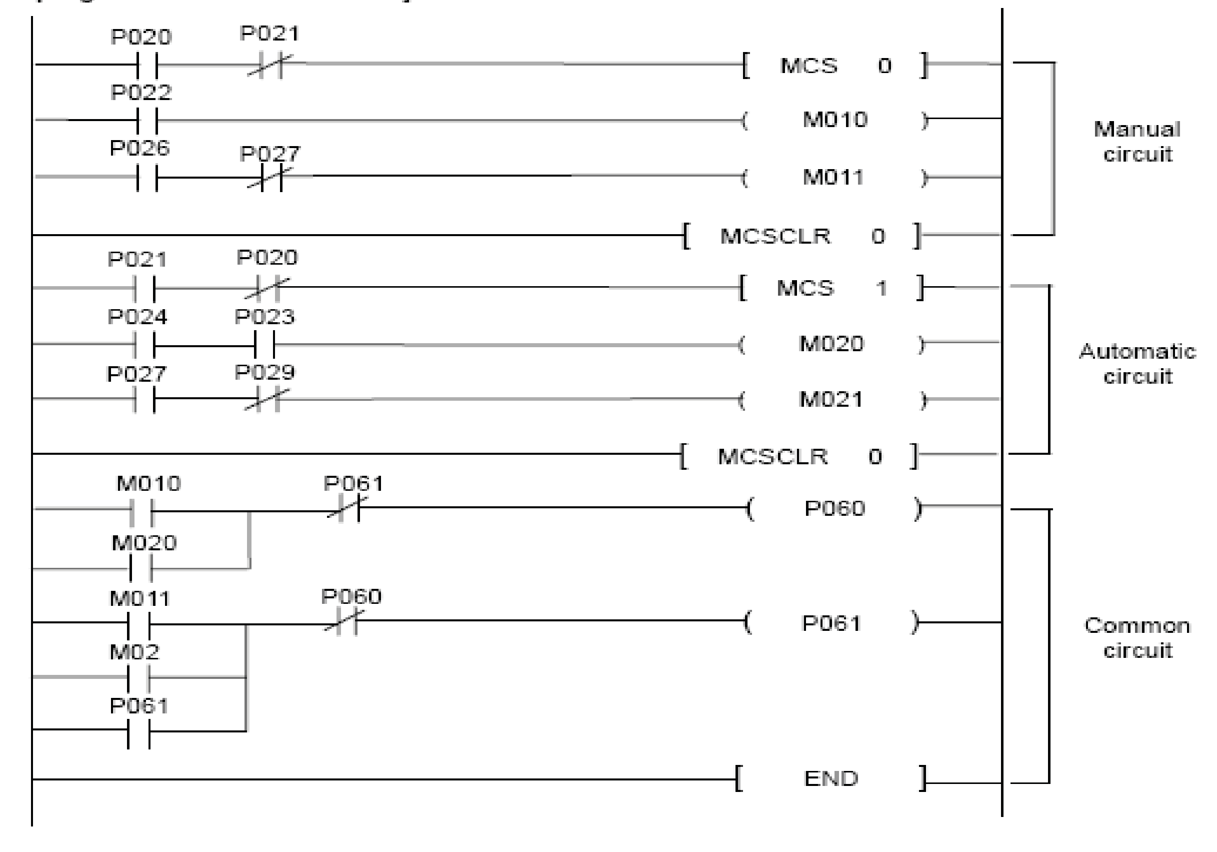
This program shows briefly an example of sequential control by using SET Sxx.xx instruction. In this example, there are ٤ processes and each process is performed in sequence. The process ٢ starts after the process ١ ended, and process ٣ starts after the process ٢ finished. When the process ٤ is completed, the process ١ will start again



The circuit with common line (Example of MCS, MCSCLR instructions) : -

The below relay circuit can not be programmed into PLC program directly. Therefore, it should be programmed with master control. (MCS and MCSCLR instructions)





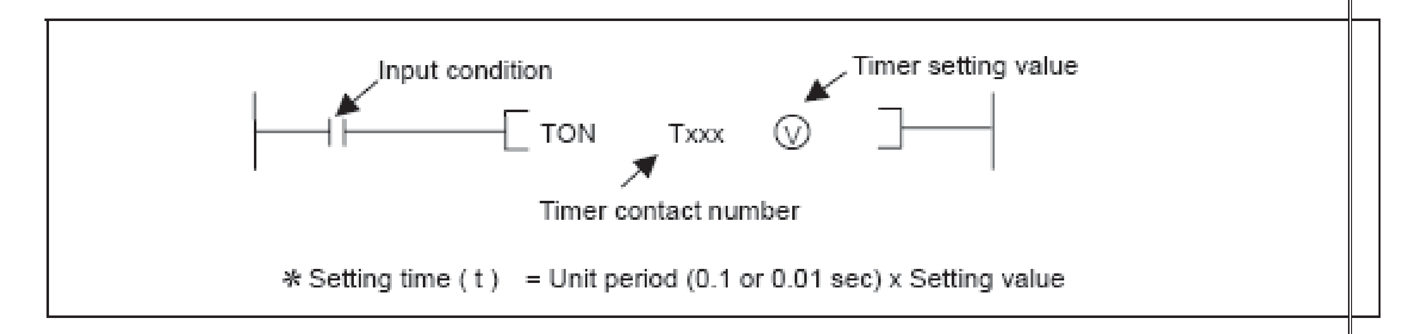
Timer/Counter Command

Timer instructions :

Mnemonic symbol	Function No.	Ladder symbol	Unit	Contents of processing
TON	-	Timer setting value TON Txxx (v) Timer relay No.	-	<On delay timer> Input t Output t = setting value
TOFF	-	Timer setting value TOFF Txxx (v) Timer relay No.	-	<Off delay timer> Input t Output t = setting value
TMR	-	Timer setting value TMR Txxx (v) Timer relay No.	-	<Accumulation timer> Input t1 t2 Output t = setting value (t = t1+t2)
TMON	-	Timer setting value TMON Txxx (v) Timer relay No.	-	<Monostable timer> Input t Output t = setting value
TRTG	-	Timer setting value TRTG Txxx (v) Timer relay No.	-	<Retriggerable timer> Input t Output t = setting value



TON On-delay timer

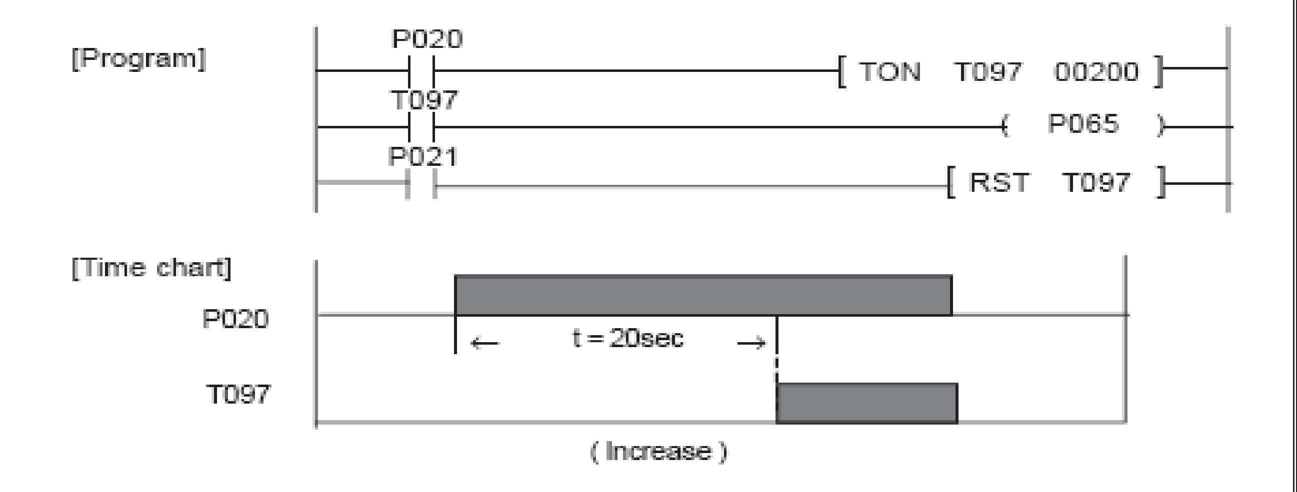


Functions :

- **A timer consists of timer contact, current value, and setting value.**
- **The current value will start to increase when the input condition turns on.**
- **It will increase by ١ at every ٠.١ or ٠.٠١ sec until it reaches to the setting value or input condition turns off.**
- **The timer contact will be switched on when the current value reaches to the setting value.**
- **The timer contact and current value is cleared when the input condition turns off or RST instruction is executed.**

Program Example :

The T٠٩٧ (٠.٠١ sec timer) will turn on ٢٠ seconds later until the P٠٢٠ is switched on.



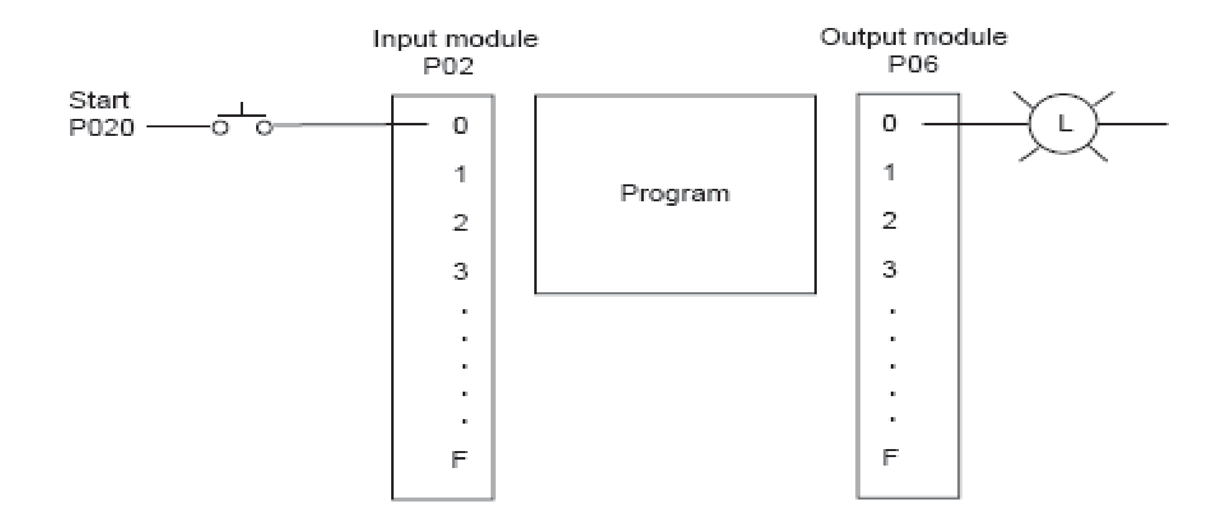
Example no. ١

A flickering lamp (example of TON instruction)

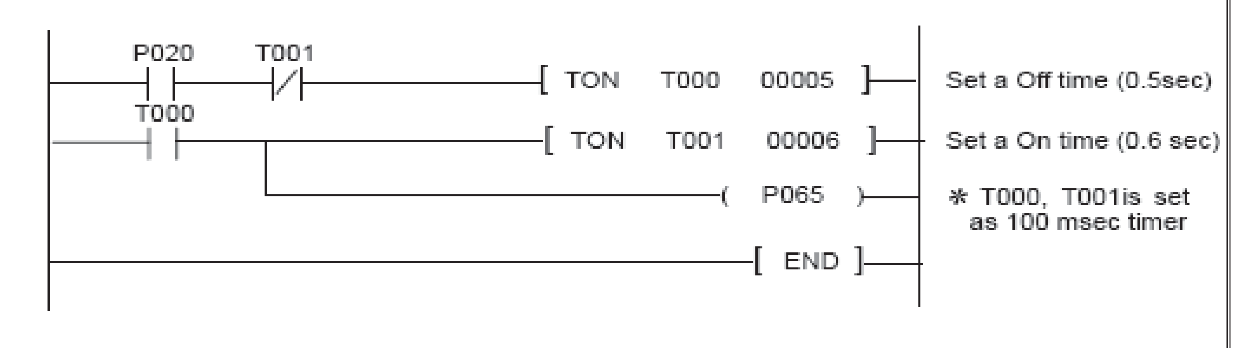
Operation :

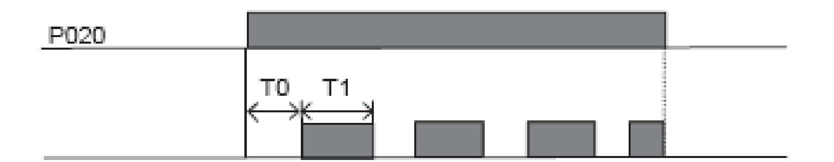
By using two timers, a lamp flickers periodically while the P٠٢٠ is on.

System diagram :

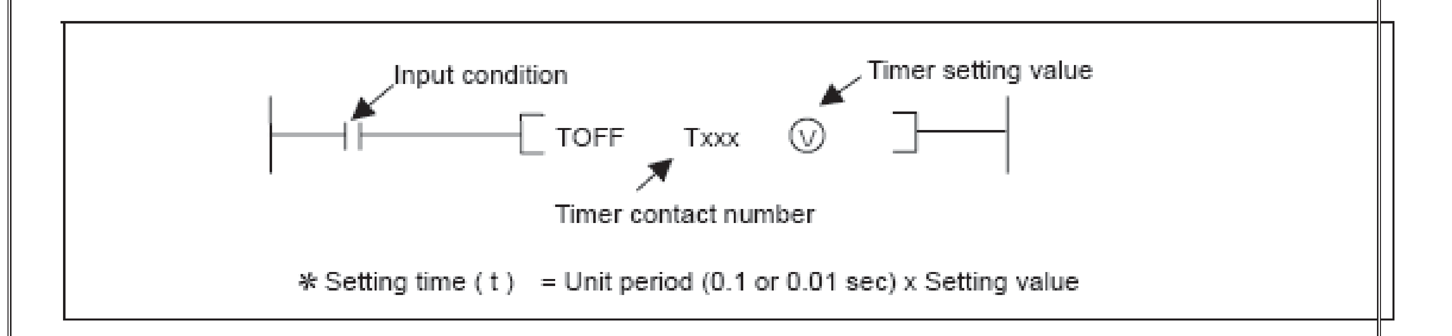


Program :





TOFF Off-delay timer

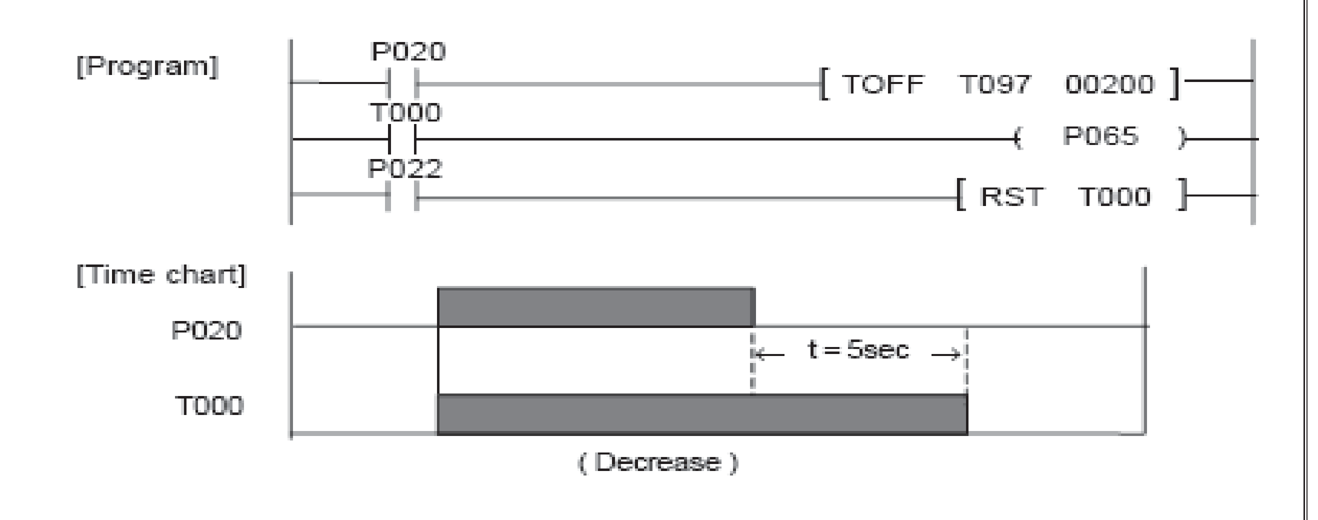


Functions :

- **A timer consists of timer contact, current value, and setting value.**
- **When the input condition turns on, the current value will be set as the setting value and the timer contact will turn on.**
- **When the input condition turns off, the current value will decrease by ١ at every ٠.١ or ٠.٠١ sec until it reaches to ٠ or input condition turns off.**
- **The timer contact will be switched off when the current value reaches to ٠.**
- **When the input condition turns off or RST instruction is executed, the timer contact will turn off and the current value will be cleared as ٠.**

Program Example :

The T٠٠٠ (٠.١ sec timer) will turn off ٥ seconds later until the P٠٢٠ is switched off.



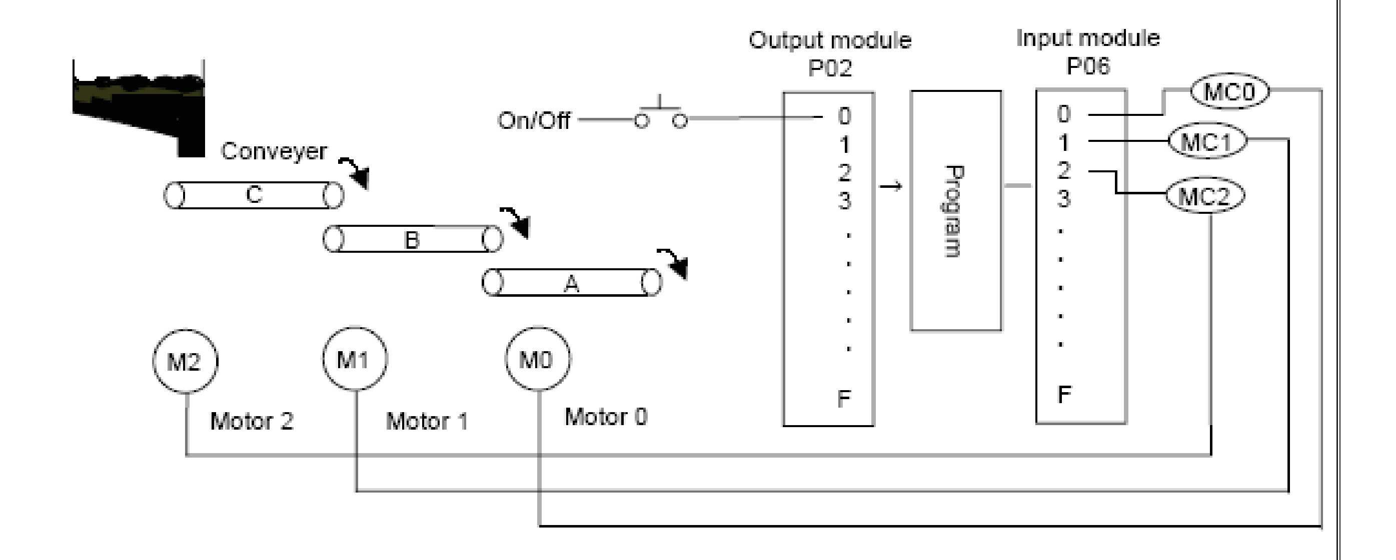
Example no. ٢

A conveyer control (example of TOFF instruction)

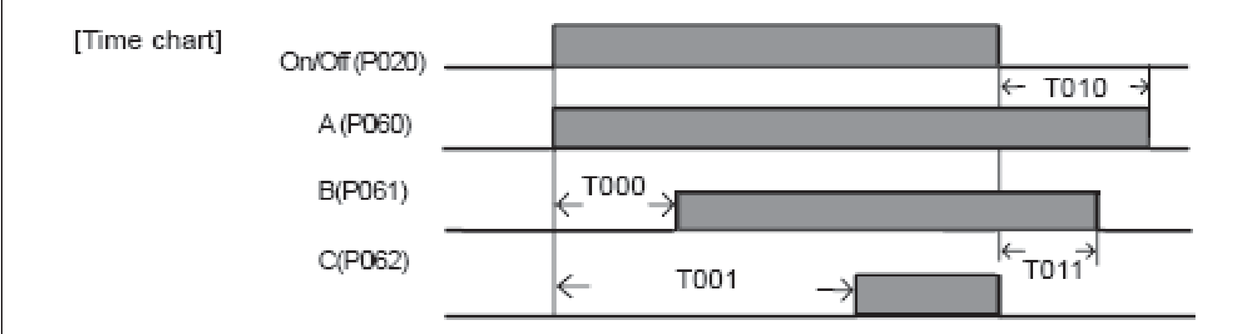
Operation :

Operate three conveyers (A, B, C) in sequence by using TOFF timers.
(Start : A – B – C, Stop : C – B – A)

System diagram :



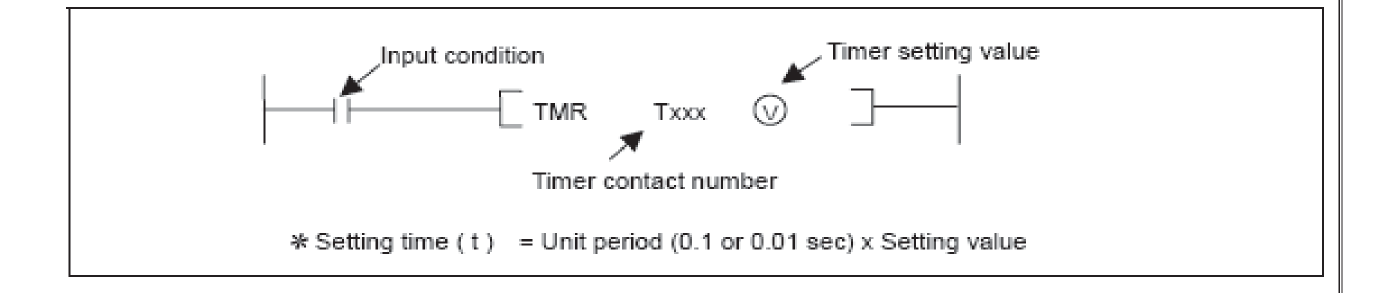
Time chart :



Program :

Step	Contact	Instruction	Comment
0000	P020	[TOFF T010 00100]	Set the off delay time of conveyer A (10sec)
0004	T010	(P060)	
0006	P020	[TON T000 00050]	Set the on delay time of conveyer B (5sec)
0010	T000	[TOFF T011 00050]	Set the off delay time of conveyer B (5sec)
0014	T011	(P062)	
0017	P020	[TON T001 00100]	Set the on delay time of conveyer C (5sec)
0021	T001	(P062)	
0024		[END]	

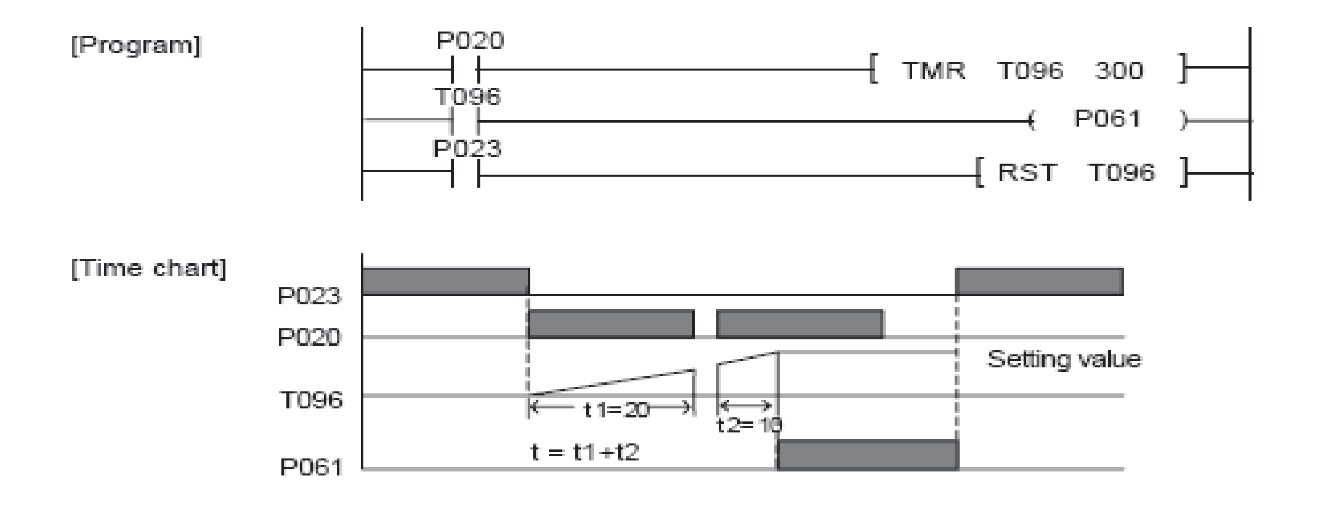
TMR Integrating timer



Functions :

- **The current value will increase by ١ while the input condition is on.**
- **When the current value reached to the setting value, the timer contact turns on.**
- **Even if the input condition is off, the current value is not cleared.**
- **If uses a timer of retentive data area, the timer will keep the current value while the CPU is powered off.**
- **When the RST instruction is executed, the timer contact and current value will be cleared as ٠.**

Program example :



Example no. ٣

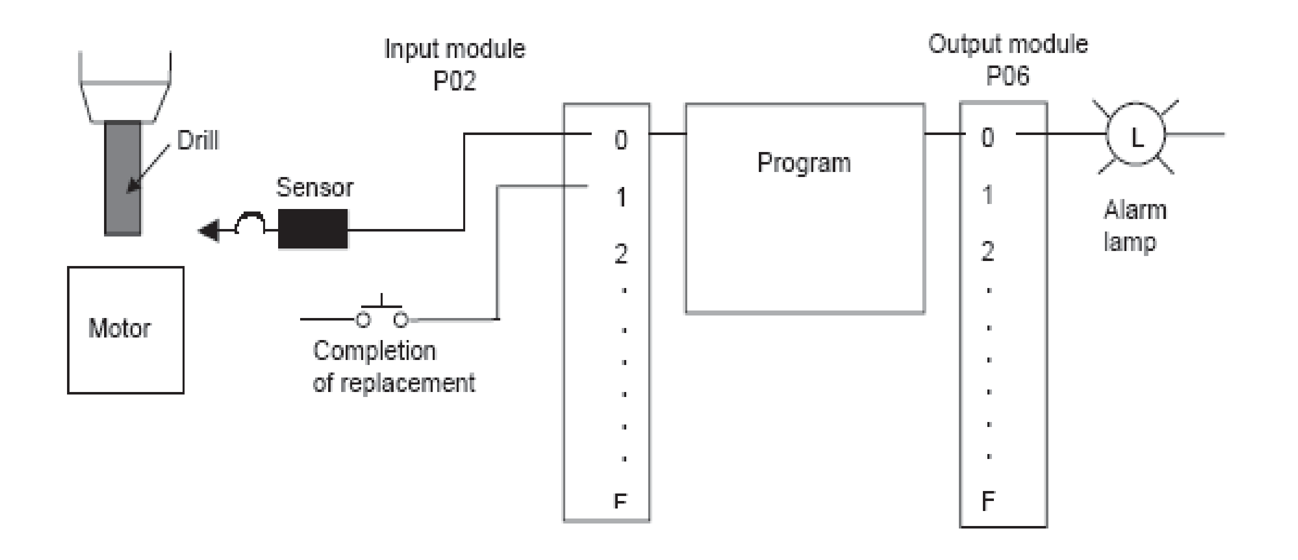
An alarm of drill replacement (example of TMR instruction)

Operation :

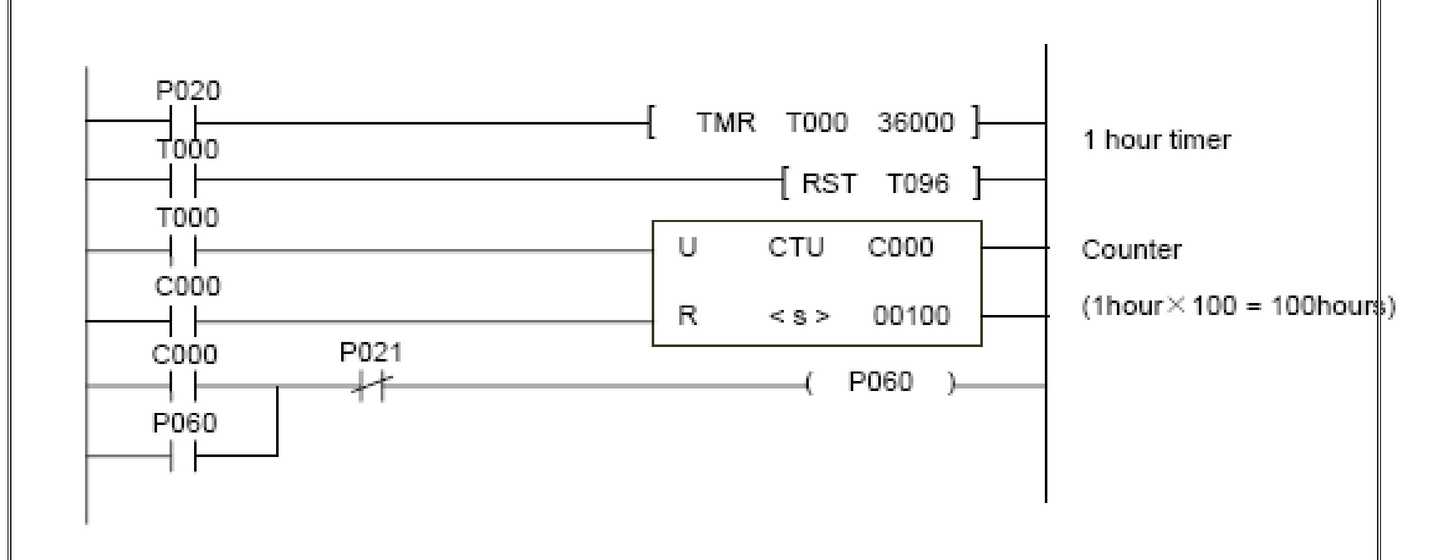
The total usage time of the drill of a machining center is counted by PLC. If the total usage time exceeds the lifetime of drill (١٠٠ hours), the PLC outputs an alarm signal to notice that a replacement of drill is required.

System diagram :

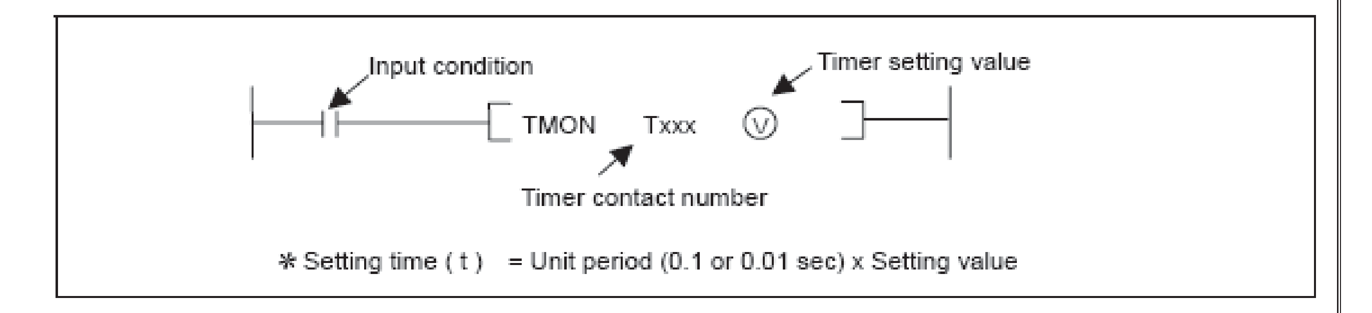
I/O	Description
P020	Detect of drill down
P021	Replacement completion
P060	Turn on an alarm lamp
T000	Timer for the lifetime of drill


Input module
P02
Output module
P06
Drill
Sensor
Program
Alarm
lamp
Motor
Completion
of replacement
0
1
2
F
L

Program :


P020
TMR T000 36000
1 hour timer
T000
RST T096
T000
U CTU C000
Counter
C000
R < s > 00100
(1hour × 100 = 100hours)
C000
P021
P060
P060

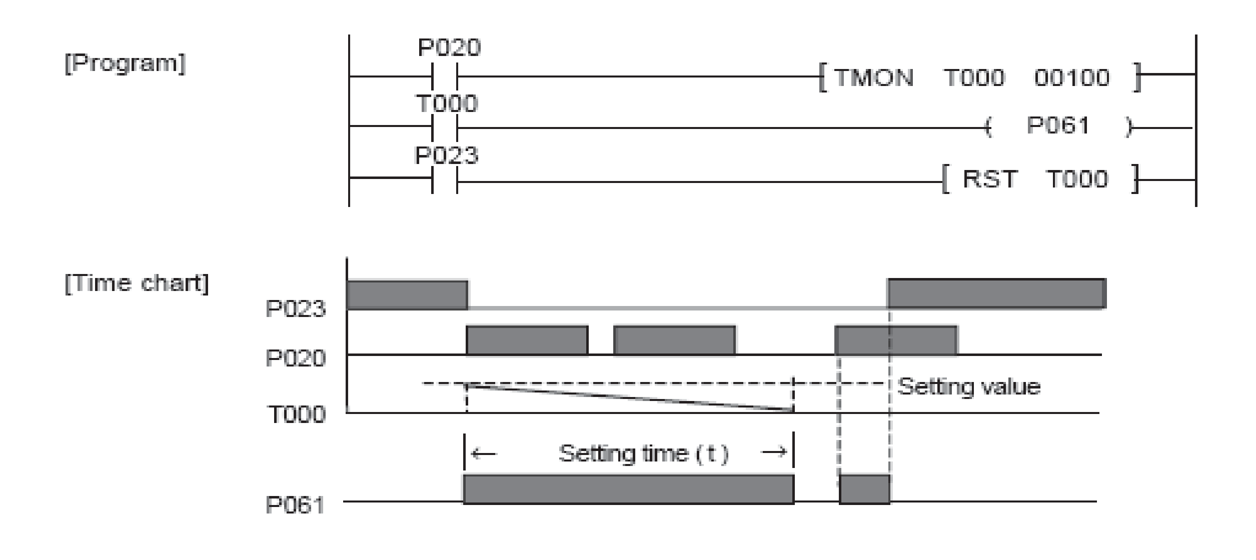
TMON Monostable timer



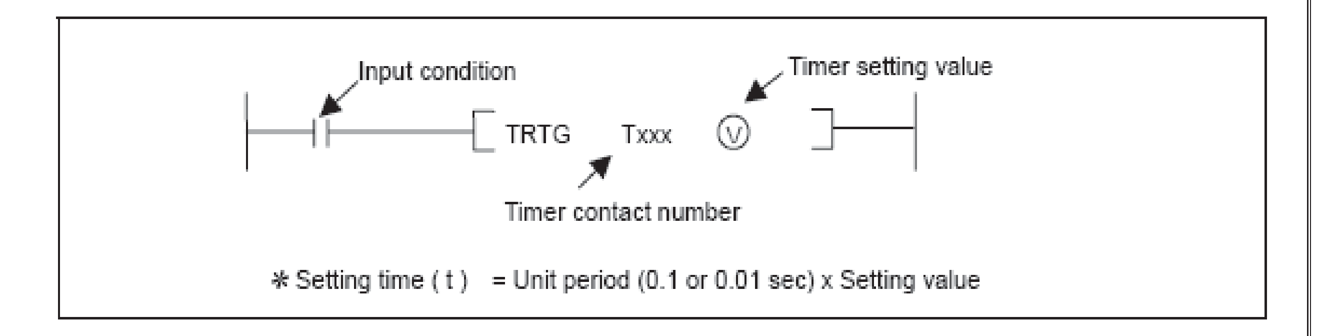
Functions :

- **When the input condition turns on, the current value will be set as the setting value and starts to decrease.**
- **The timer contact turns on when the input condition turns on.**
- **When the input condition turns off, the current value will decrease by ١ at every ٠.١ or ٠.٠١ sec until it reaches to ٠ and the timer contact will be switched off when the current value reaches to ٠.**
- **While a timer is operating, on/off changed of input condition is ignored.**
- **When the RST instruction is executed, the timer contact will turn off and the current value will be cleared as ٠.**

Program Example :



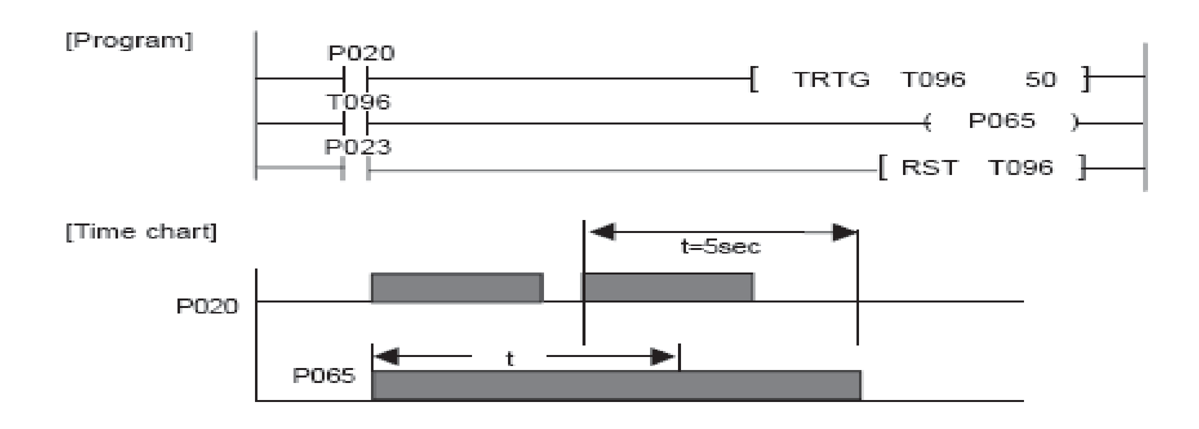
TRTG Retrigerrable timer



Functions :

- **When the input condition turns on, the current value will be set as the setting value and starts to decrease.**
- **The timer contact turns on when the input condition is switched on.**
- **The current value will decrease by ١ at every ٠.١ or ٠.٠١sec until it reaches to ٠ and the timer contact will be switched off when the current value reaches to ٠.**
- **If the input condition turns on again during timer operation, the current value will reset as the setting value and re-start to decreasing from the setting value.**
- **When the RST instruction is executed, the timer contact will turn off and the current value will be cleared as ٠.**

Program example :



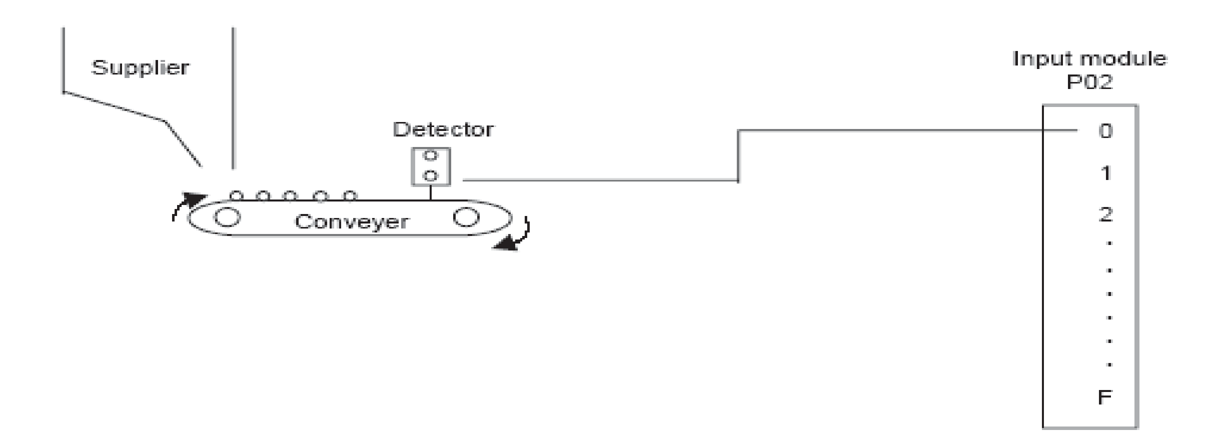
Example no. ٤

The fault of conveyer detecting circuit (example of TRTG instruction)

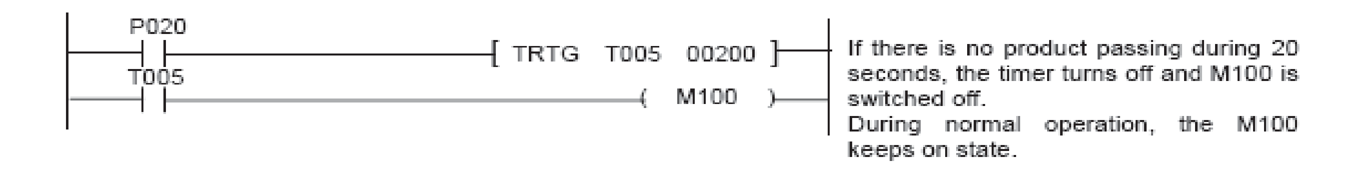
Operation :

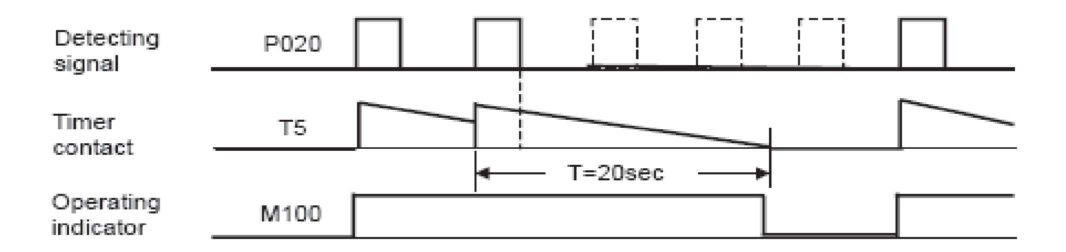
Detect the fault of conveyer by check that a product is passed within a specified period or not.

System diagram :



Program :

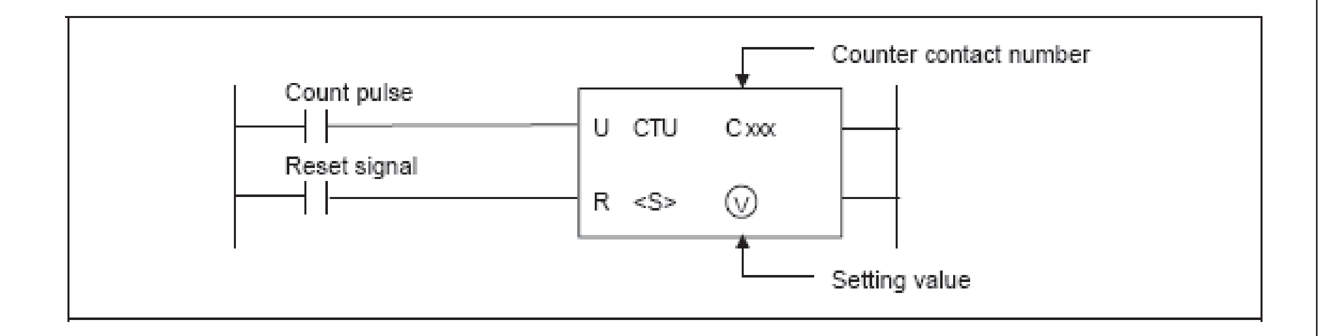




Counter instructions :

Counter instruction	Type	Counting method	Input signal	Time chart
CTU	Up Counter	Increment	1	Reset signal Count Pulse Elapsed value Counter output Setting value
CTD	Down counter	Decrement	1	Reset signal Count Pulse Elapsed value Counter output Setting value
CTUD	Up/Down Counter	Increment / Decrement	2	Reset signal Increase pulse Decrease pulse Elapsed value Counter output Setting value
CTR	Ring counter	Increment	1	Reset signal Count Pulse Elapsed value Counter output Setting value

 **CTU Up counter**

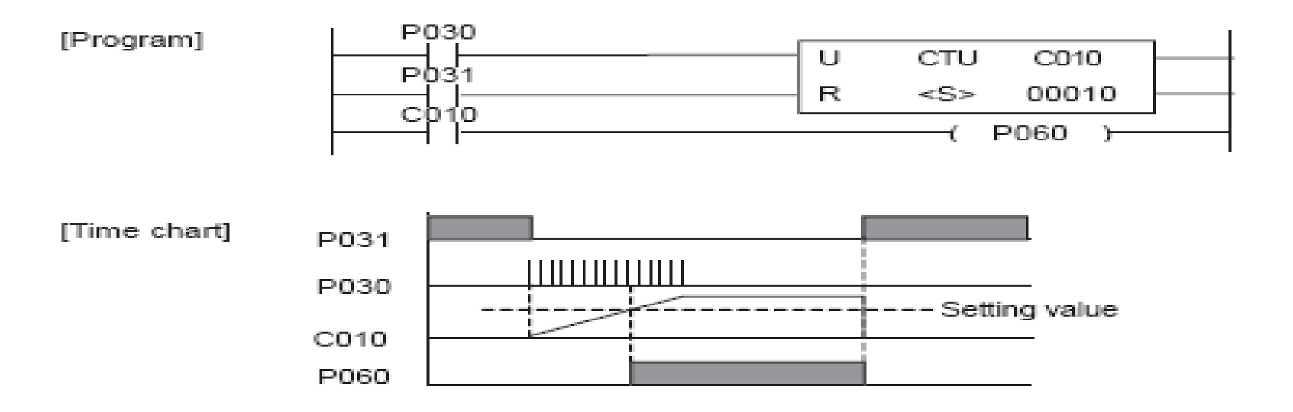


Functions :

- **Whenever a rising edged is detected at the count pulse input, the current value is increased by ١**
- **The initial current value is ٠ and when the current value is reached to the setting value, the counter contact turns on.**
- **After the counter contact turns on, the current value keeps increasing until its maximum value. (٦٥٥٣٥)**
- **When the reset signal is switched on, the counter contact and current value is cleared as ٠.**

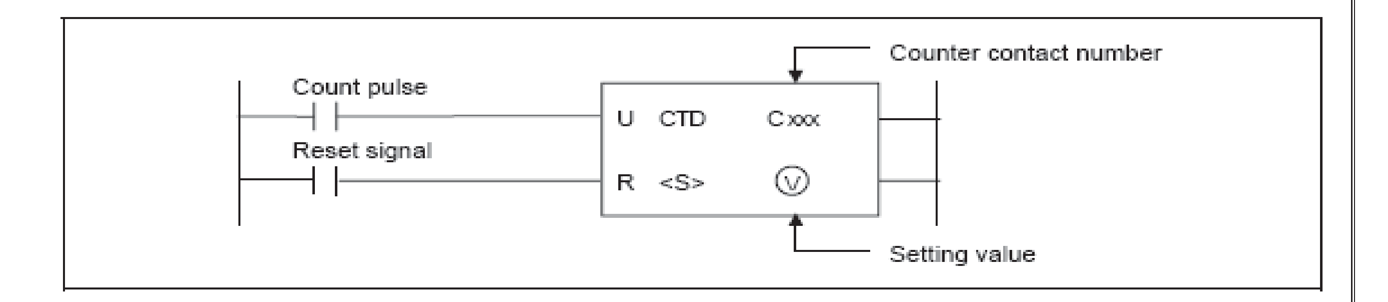
Program example :

Whenever the P٠٣٠ is changed from off to on, the current value of C٠١٠ is increased by ١. The P٠٣١ is reset condition.





CTD Down counter

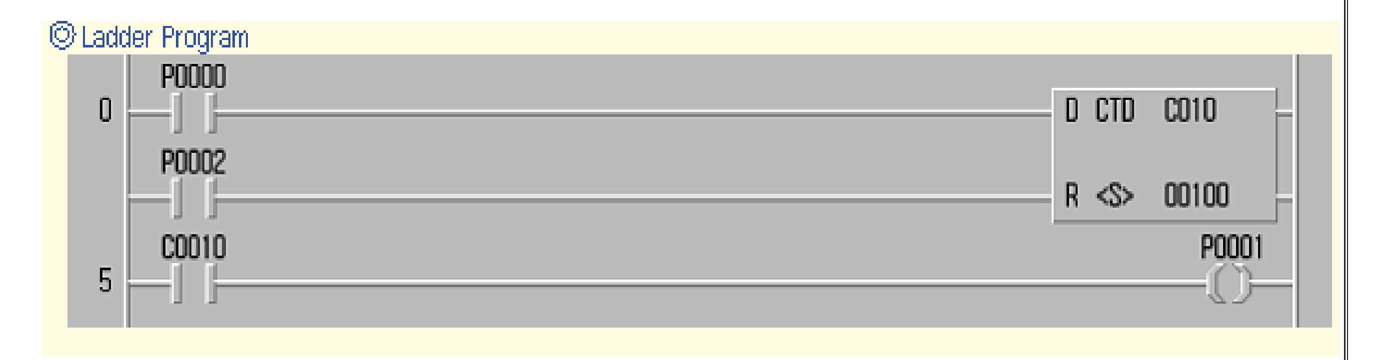


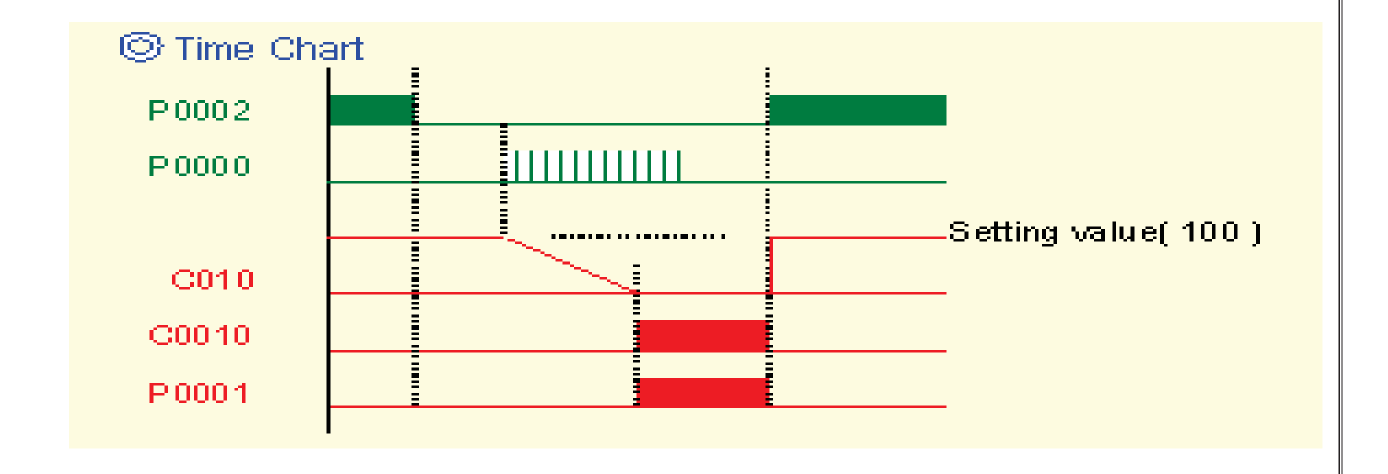
Functions :

- **Whenever the rising edge is detected from counter pulse input, the current value is decreased by ١.**
- **The initial current value is the setting value, and when the current value reached to ٠, the counter contact is switched on.**
- **When the reset signal turns on, the counter contact is switched off and the current value is reset as the setting value.**

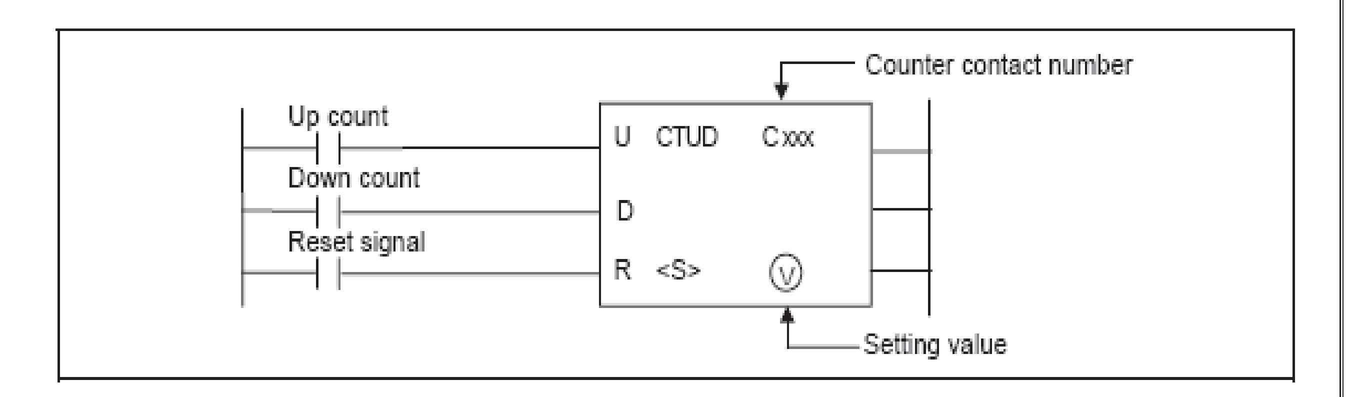
Program example :

Whenever the P٠٠٠ is changed from off to on, the current value of C٠١٠ is decreased by ١. The P٠٠٠٢ is reset condition.





CTUD Up-down counter



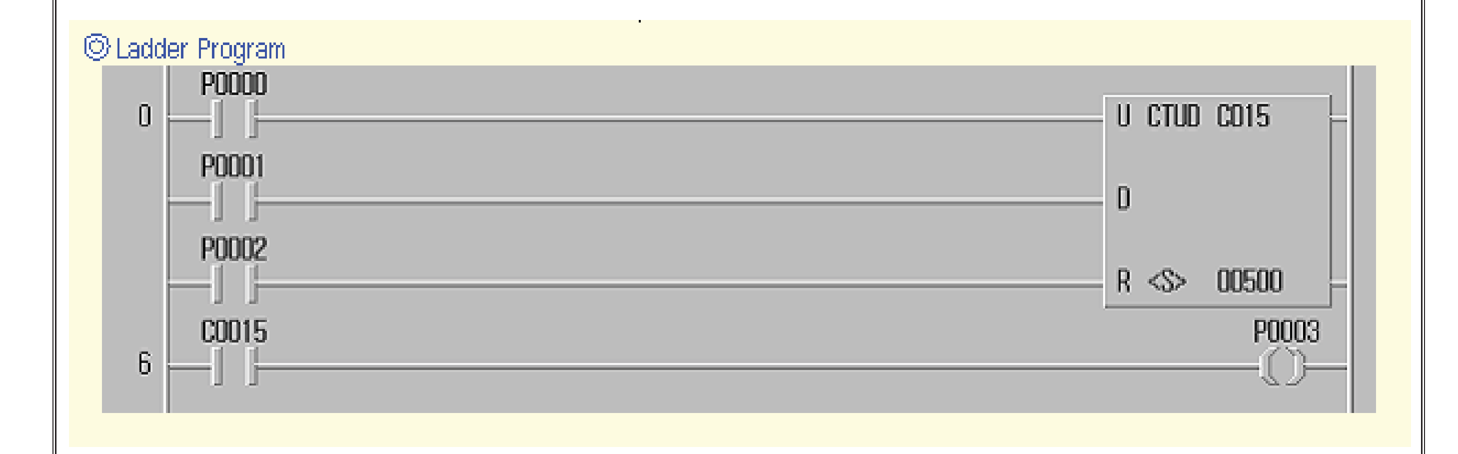
<u>Functions :</u>

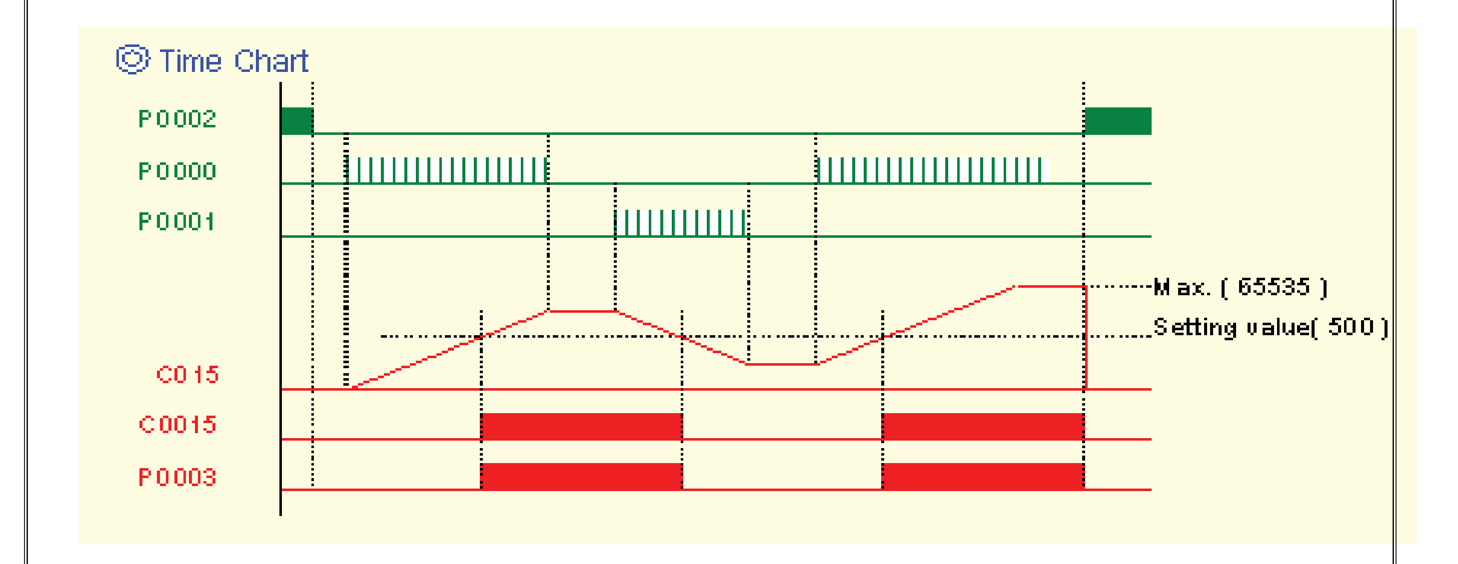
- **Whenever a rising edged is detected from up count input, the current value is increased by ١.**
- **The current value is decreased by ١ whenever a rising edge is detected at the down count input.**
- **The initial current value is ٠.**
- **The counter contact turns on when the current value is same or greater than the setting value.**
- **When the reset signal turns on, the counter contact and current value is cleared as ٠.**

Program example :

The P٠٠٠٠ is up count input, and the P٠٠٠١ is down count input.

The P٠٠٠٢ is reset signal.





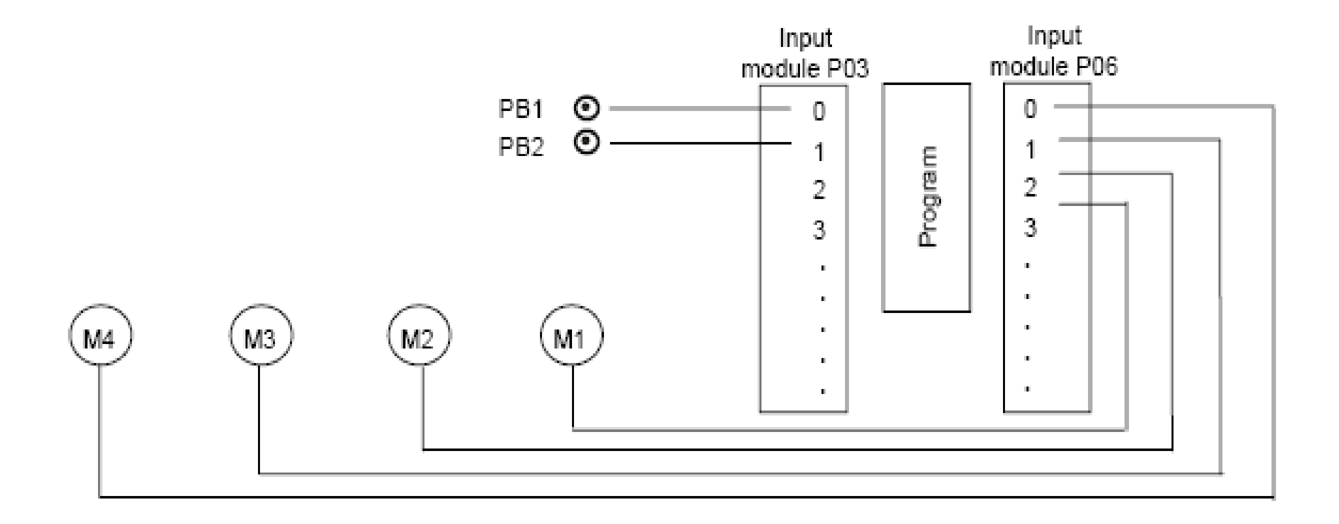
Example no. ٥

A control circuit for motor operation (example of CTUD instruction)

Operation :

There are ٤ motors controlled by PLC. Whenever the push-button PB١ is pressed, the numbers of operating motor is increased by ١. The PB٢ decreases the numbers of operating motor whenever it is pressed. If the PB١ is pushed when ٤ motors are operating, all motors will stop their operation.

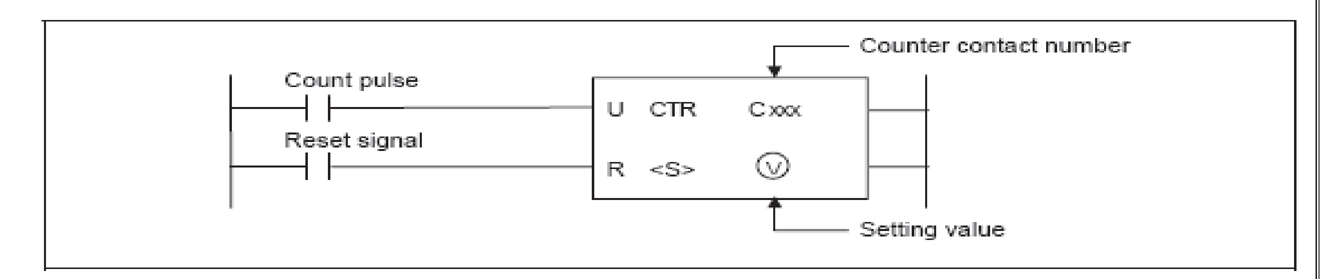
System diagram :



Program :

Try to do this program.

CTR Ring counter



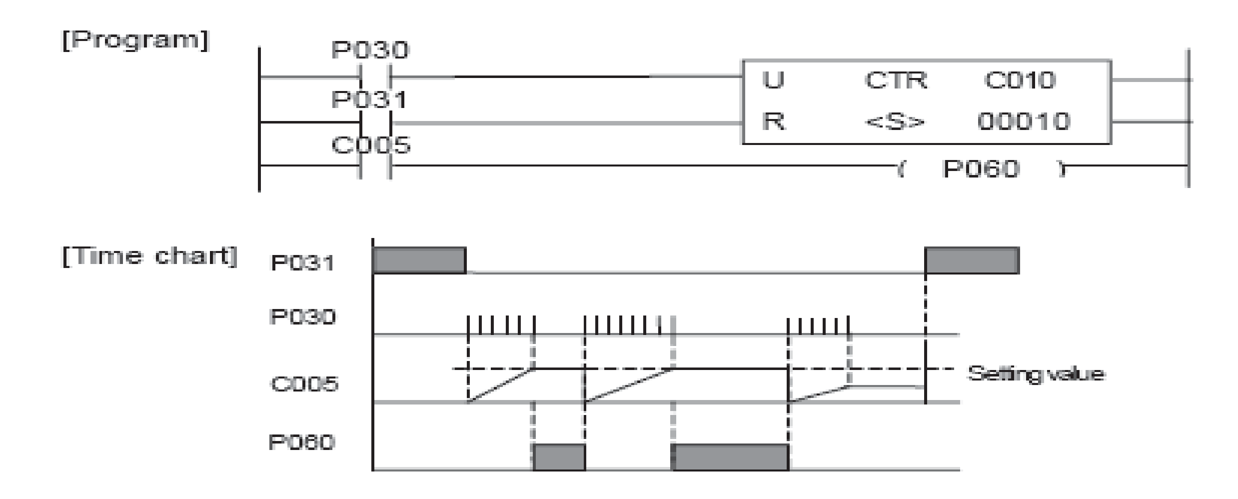
Functions :

- **Whenever a rising edge is detected at the count pulse input, the current value is increased by ١.**
- **If the current value is reached to the setting value, the counter contact is switched on. Then the counter contact and current value will be cleared as ٠ when the next rising edge is applied to the count pulse input.**
- **When the reset signal turns on, the counter contact and current value will be cleared as ٠.**

Program example :

The P٠٣٠ is count pulse input and when the current value is same as the setting value, the counter contact is switched on.

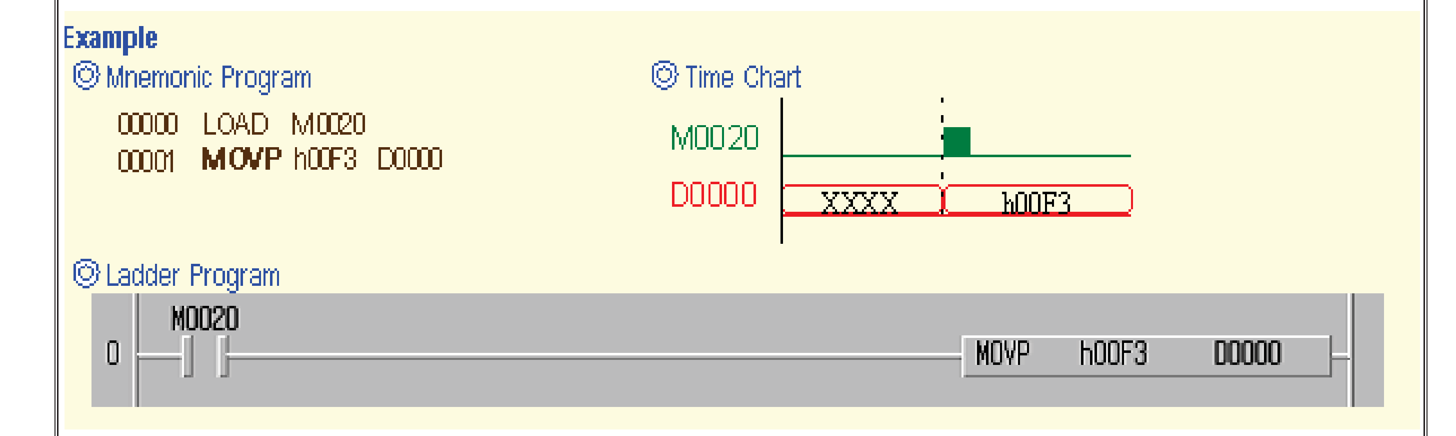
When the P٠٣٠ is switched on ١١th time, the counter contact (P٠٦٠) is off and the current value is cleared as ٠.



Transfer Command

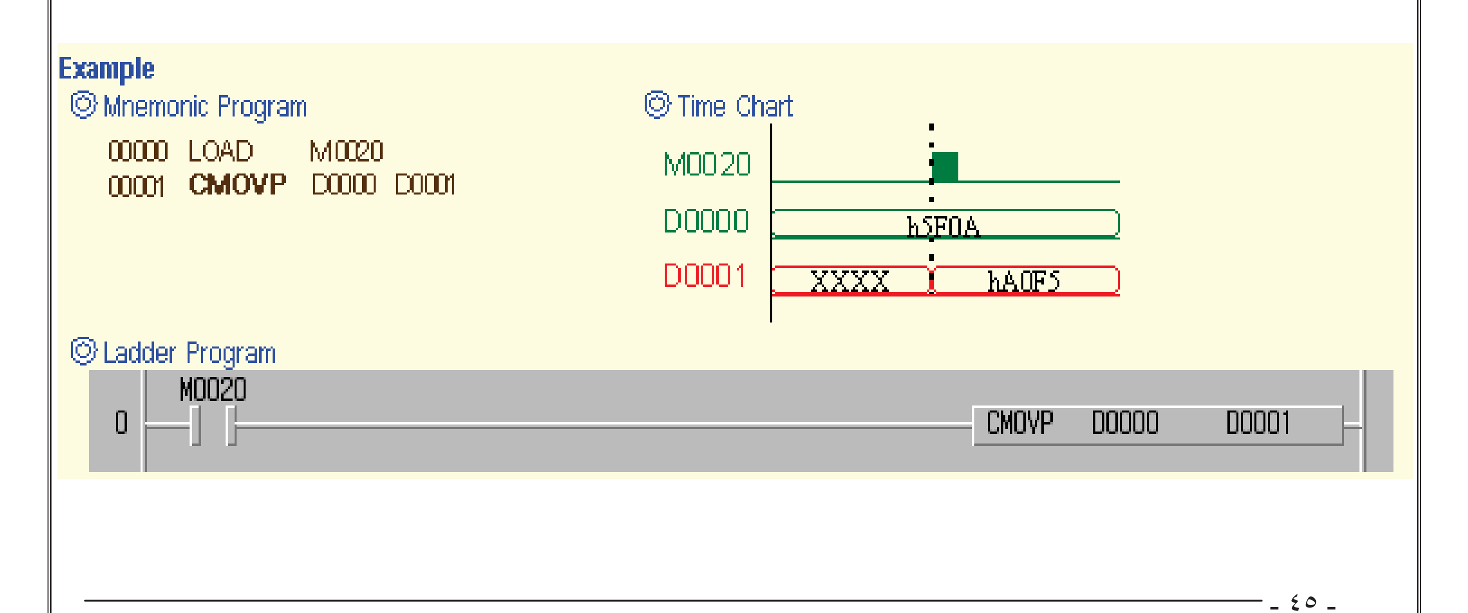
- **MOV** ٥ **Data Copy**
- **MOVP** ٥ **Data Copy (Pulse)**
- **DMOV** ٥/٧ **Data Copy (Double)**
- **DMOVP** ٥/٧ **Data Copy (Double & Pulse)**

It use for transfer Data from the specified device s and copied to the destination device d.



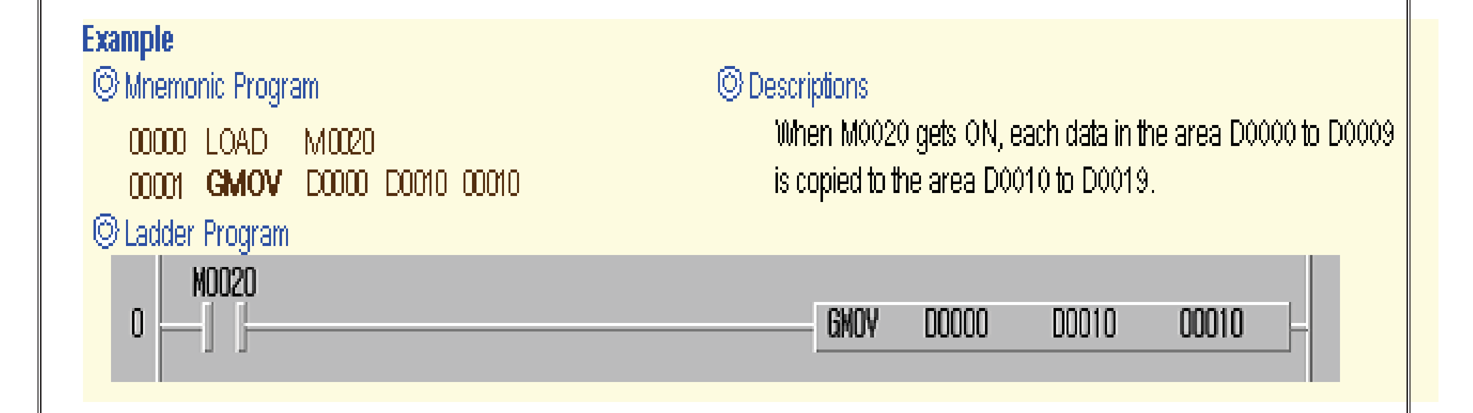
- **CMOV** ٥ **Complement Data Copy**
- **CMOVP** ٥ **Complement Data Copy (Pulse)**
- **DCMOV** ٥/٧ **Complement Data Copy (Double)**
- **DCMOVP** ٥/٧ **Complement Data Copy (Double & Pulse)**

Transfers ١'s complement of data in the specified device s to the destination device d, that is, the converted data in the device s is stored in the device d.



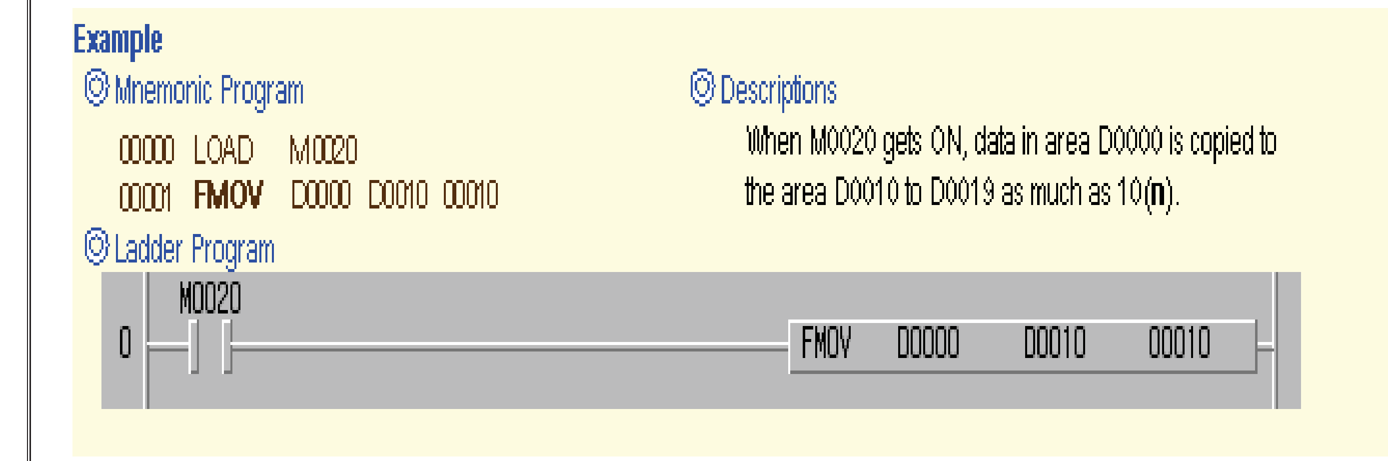
- **GMOV** ٧ **Group Data Copy**
- **GMOVP** ٧ **Group Data Copy (Pulse)**

Copies data as much as n (number of words) from the leading area of the specified device s to the leading device d as much as n .



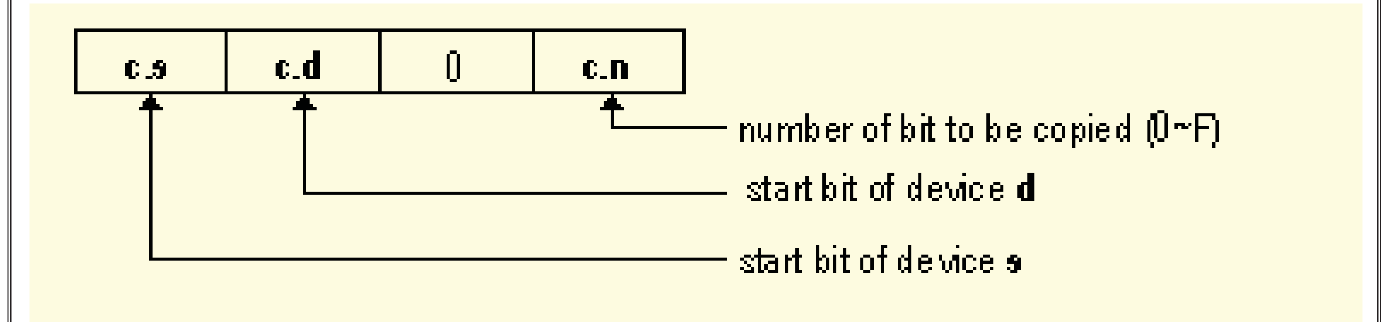
- **FMOV** ٧ **Multiple Data Copy**
- **FMOVP** ٧ **Multiple Data Copy (Pulse)**

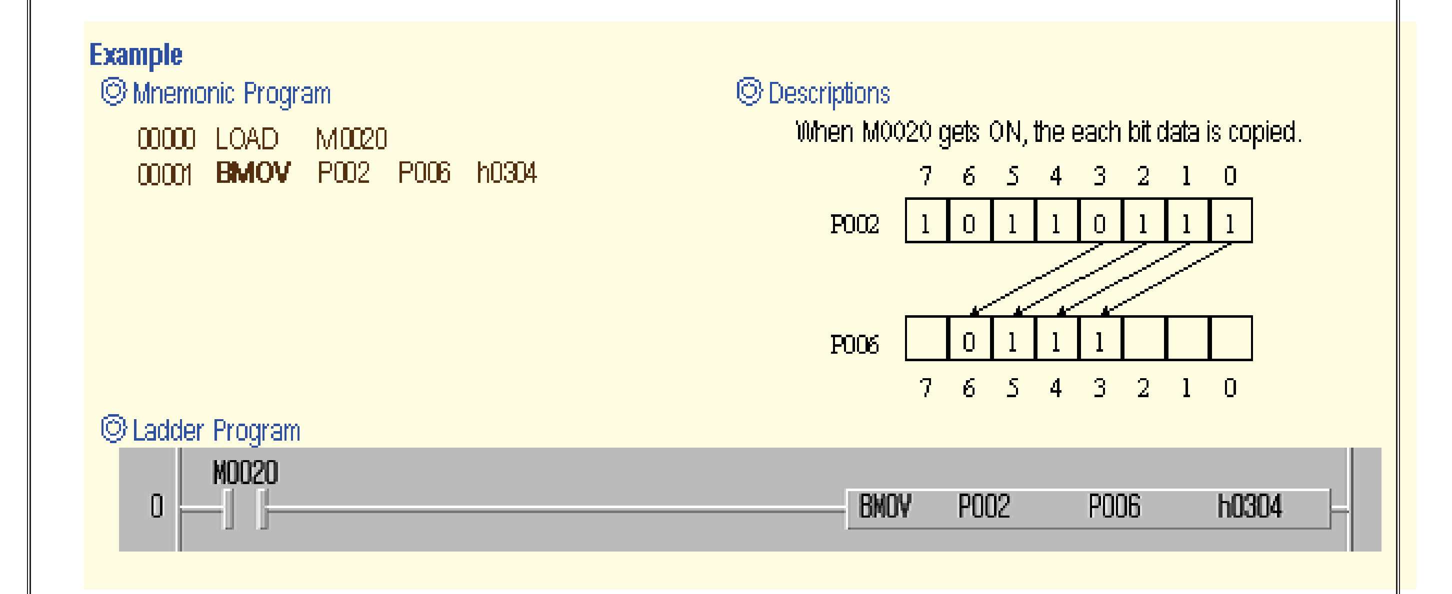
Copies data in the specified device s to the destination device d as much as n (number of words).



- **BMOV** ٧ **Bit Data Copy**
- **BMOVP** ٧ **Bit Data Copy (Pulse)**

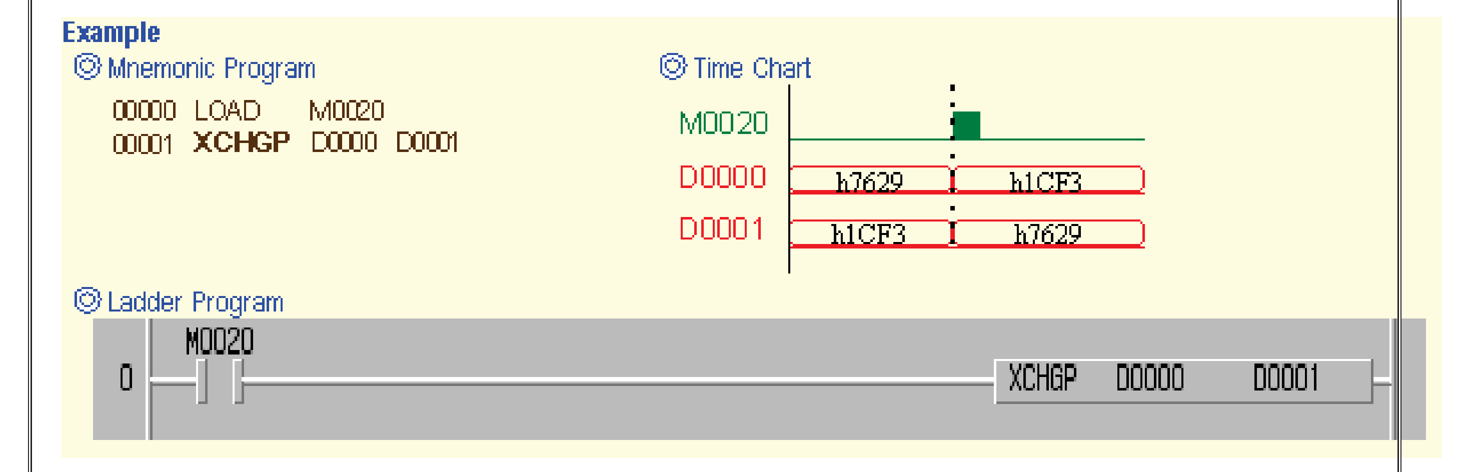
Copies the specified number of bits(c.n) from the start bit(c.s) of device s to the start bit(c.d) from the device d as much as the specified number of bits(c.n).





- **XCHG** ⚬ **Data Exchange**
- **XCHGP** ⚬ **Data Exchange (Pulse)**
- **DXCHG** ⚬ **Data Exchange (Double)**
- **DXCHGP** ⚬ **Data Exchange (Double & Pulse)**

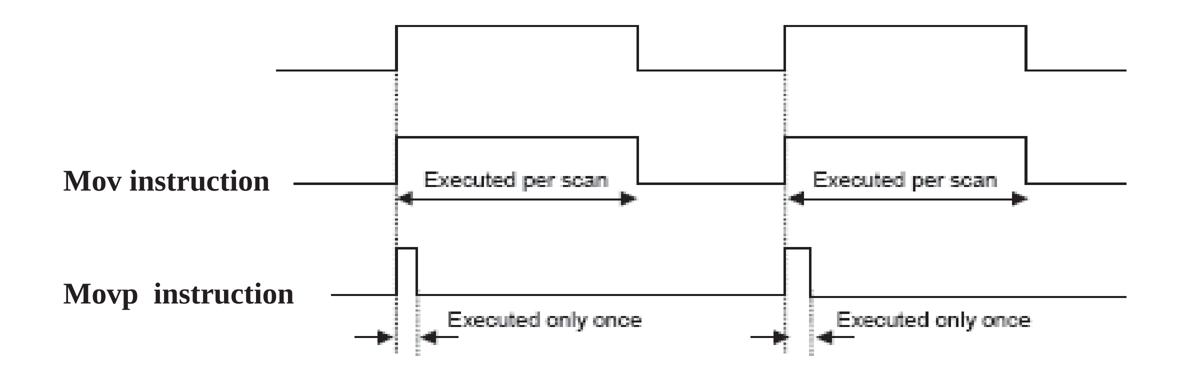
Exchanges the data in the device d١ with the data in the device d٢.



<u>Note that : -</u>

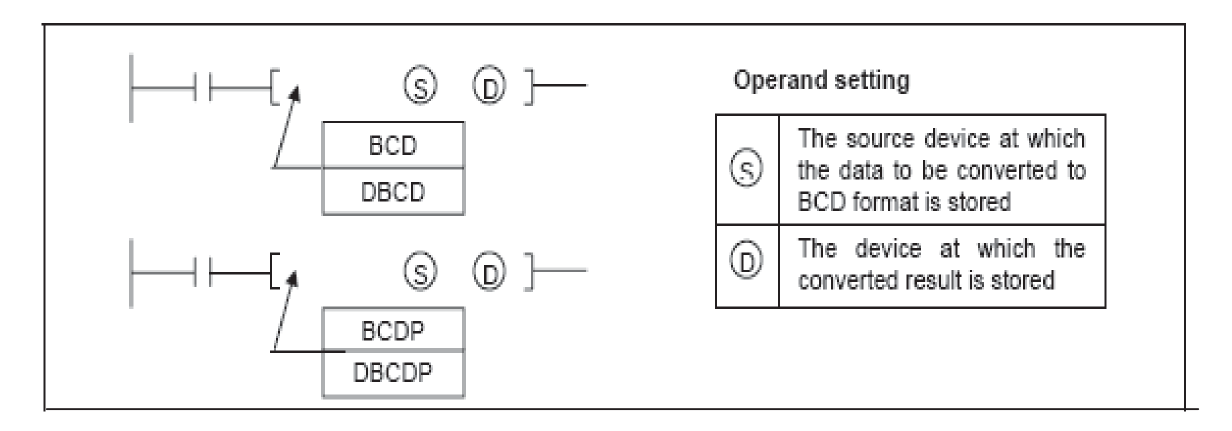
When the instruction end with (P) it mean that the instruction is executed once at every rising edge of the input .

When the instruction end without (P) it mean that the instruction is executed every scan of the input .



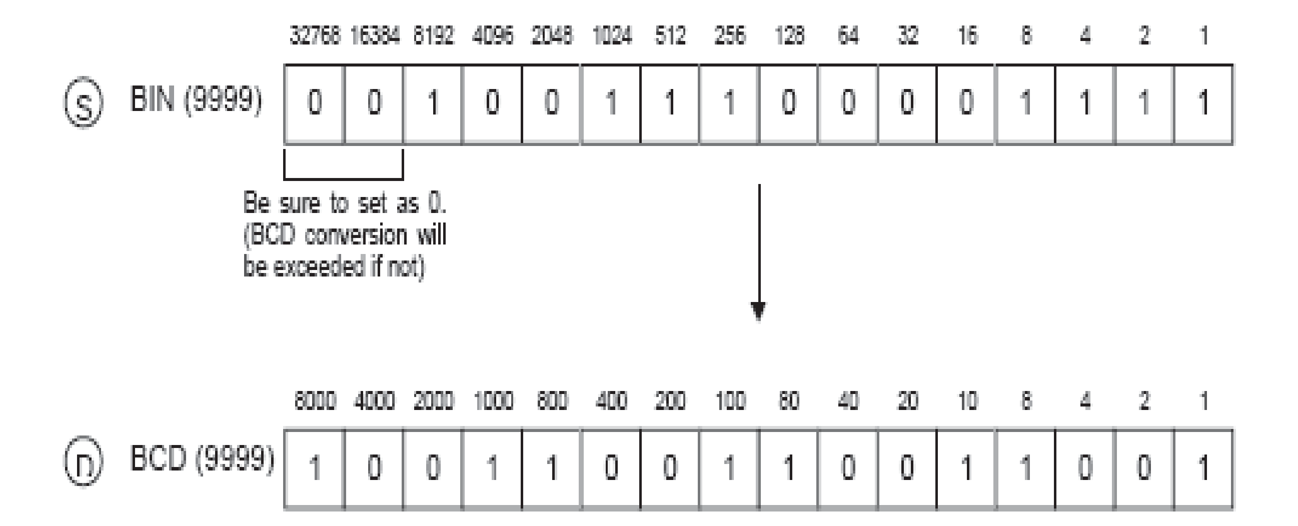
Conversion Command

- **BCD** Converts binary data into BCD data
- **BCDP** Converts binary data into BCD data (Pulse)
- **DBCD** Converts binary data into BCD data (Double)
- **DBCDP** Converts binary data into BCD data (Double & Pulse)



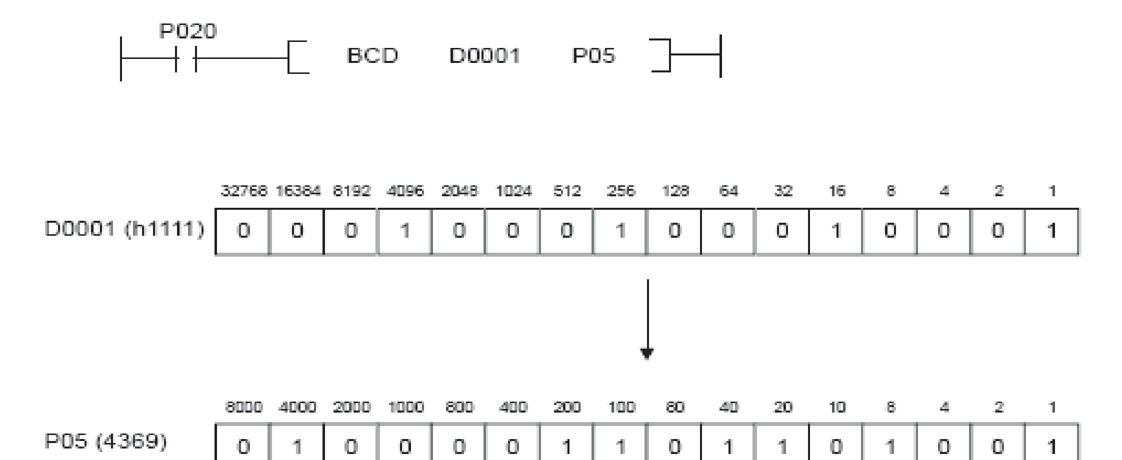
١) Functions

BCD : Converts binary data (٠ to ٩٩٩٩) of the device specified at [S] into BCD format and transfers the result to the device specified at [D].



٢) Program example : -

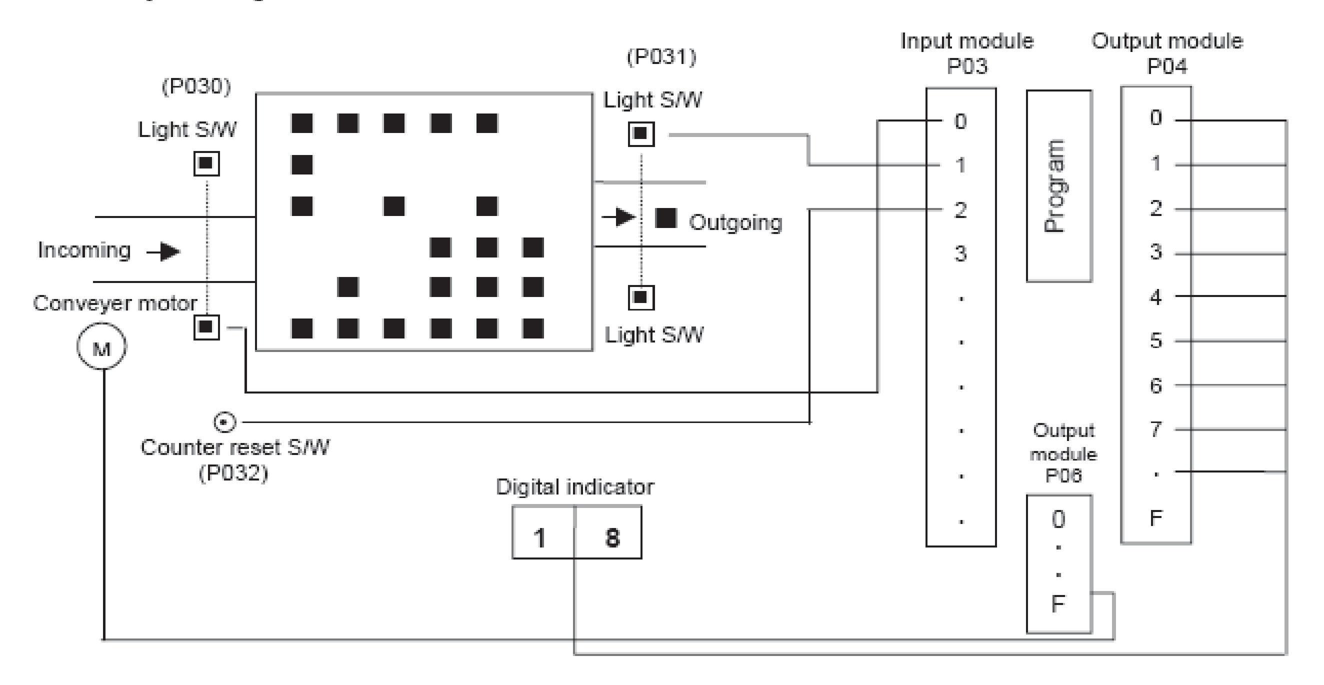
While P٠٢٠ is on, convert the binary data of D٠٠١ and transfer the result to the P٠٥ word.



Displaying the current value of counter
(example of BCD, BMOV instructions)

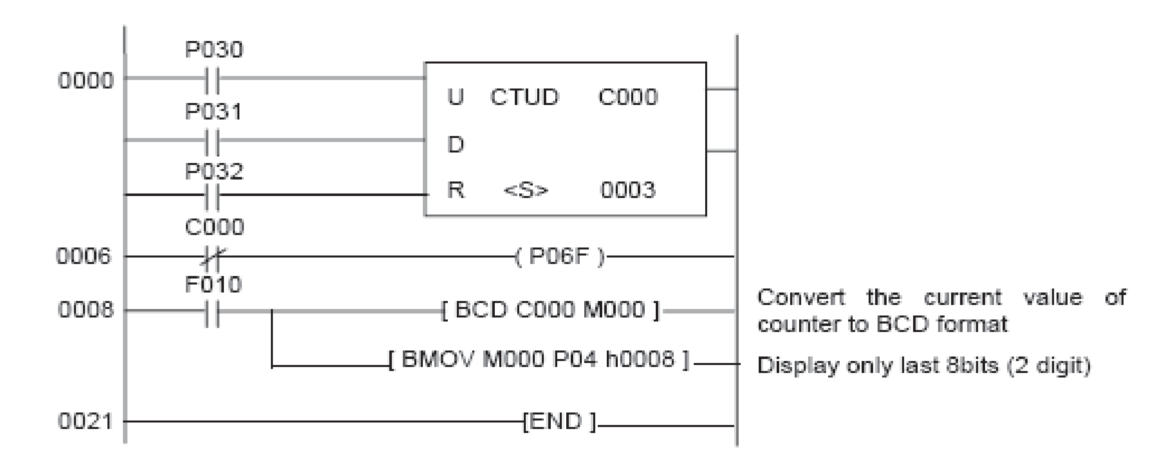
١. **Operation There is a warehouse and numbers of product incoming / outgoing are counted by light switches. The current stock of inside of warehouse is displayed by a digital indicator. When the stock of inside of warehouse reaches to ٣٠, the incoming**

2. System diagram



٢. **conveyer will be stopped.**

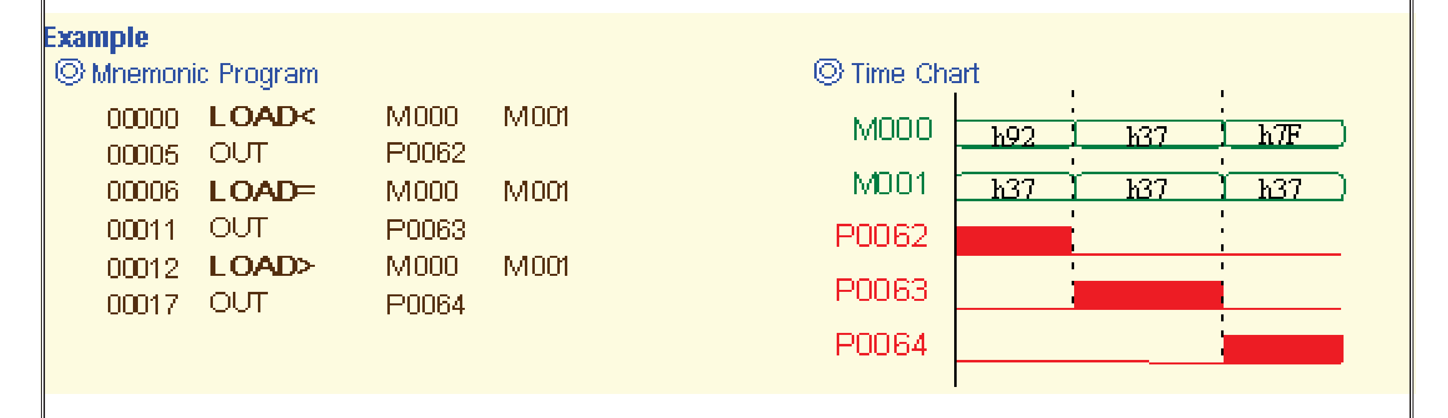
3. Program

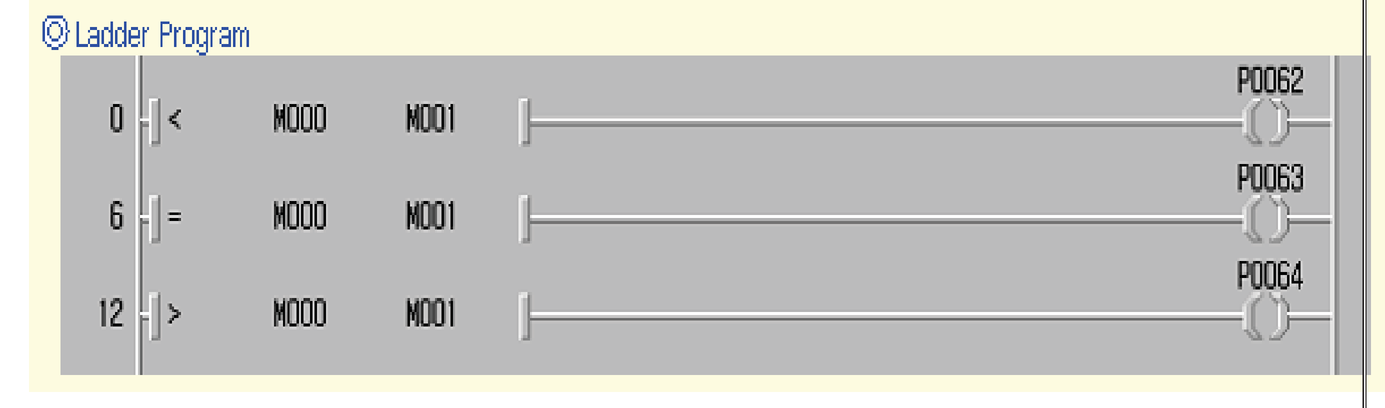


Comparison Command

Command	Step No.	Function
LOAD=	٥	LOAD Comparison Result(=)
LOAD<>	٥	LOAD Comparison Result (<>)
LOAD>=	٥	LOAD Comparison Result (>=)
LOAD<=	٥	LOAD Comparison Result (<=)
LOAD>	٥	LOAD Comparison Result (>)
LOAD<	٥	LOAD Comparison Result (<)
LOADD=	٥/٧/٩	LOAD Comparison Result (=) (Double)
LOADD<>	٥/٧/٩	LOAD Comparison Result (<>) (Double)
LOADD>=	٥/٧/٩	LOAD Comparison Result (>=) (Double)
LOADD<=	٥/٧/٩	LOAD Comparison Result (<=) (Double)
LOADD>	٥/٧/٩	LOAD Comparison Result (>) (Double)
LOADD<	٥/٧/٩	LOAD Comparison Result (<) (Double)

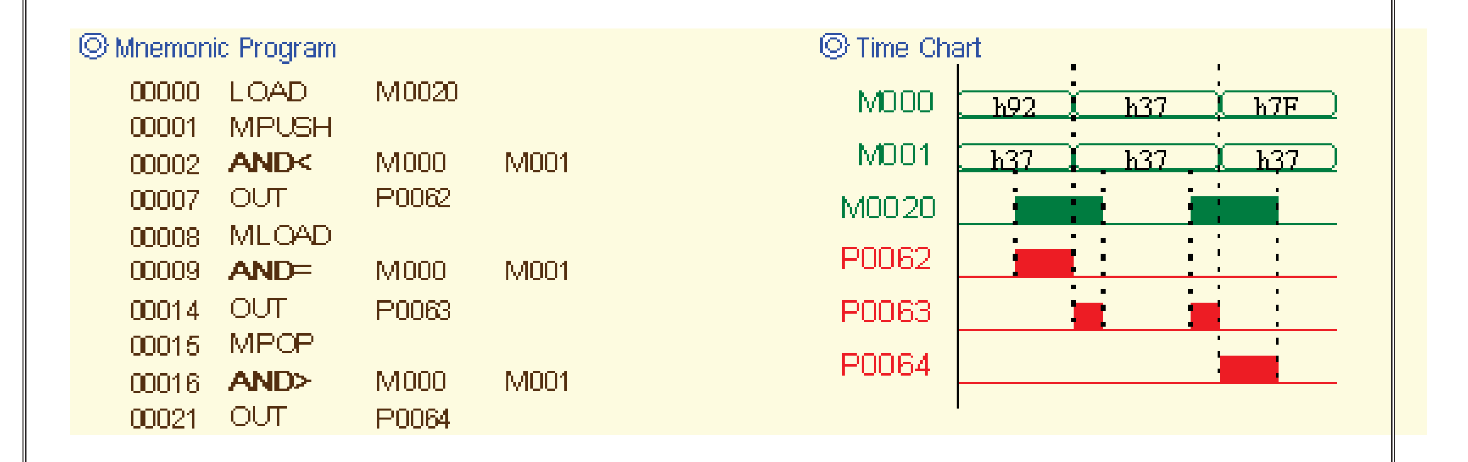
If the result of comparison of s١ and s٢ is satisfied with the condition, the result gets ON. If not, the result gets OFF.

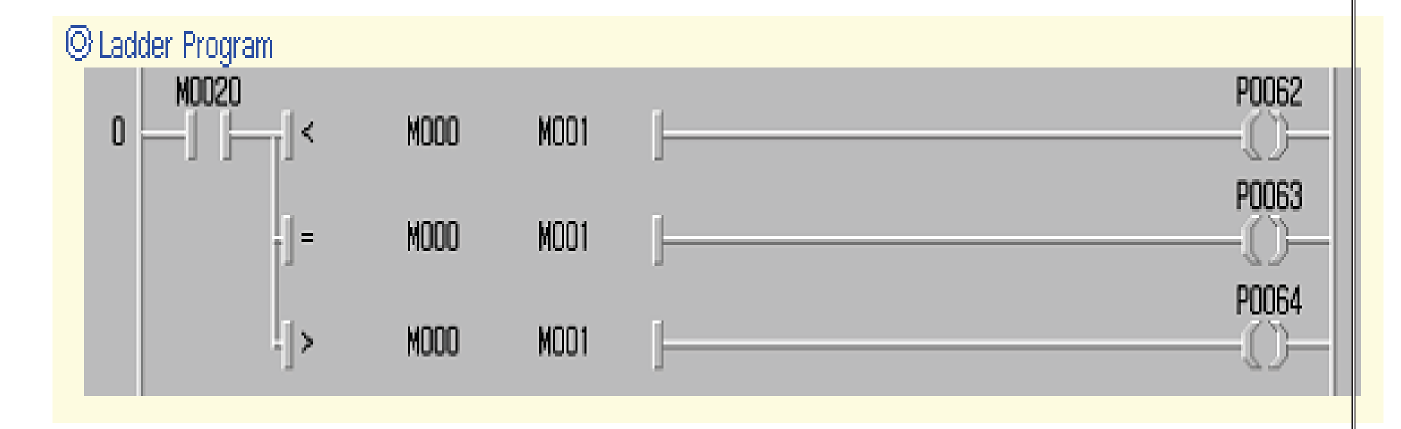




- **AND=** ٥ **AND Comparison Result (=)**
- **AND<>** ٥ **AND Comparison Result (<>)**
- **AND>=** ٥ **AND Comparison Result (>=)**
- **AND<=** ٥ **AND Comparison Result (<=)**
- **AND>** ٥ **AND Comparison Result (>)**
- **AND<** ٥ **AND Comparison Result (<)**
- **ANDD=** ٥/٧/٩ **AND Comparison Result (=) (Double)**
- **ANDD<>** ٥/٧/٩ **AND Comparison Result (<>) (Double)**
- **ANDD>=** ٥/٧/٩ **AND Comparison Result (>=) (Double)**
- **ANDD<=** ٥/٧/٩ **AND Comparison Result (<=) (Double)**
- **ANDD>** ٥/٧/٩ **AND Comparison Result (>) (Double)**
- **ANDD<** ٥/٧/٩ **AND Comparison Result (<) (Double)**

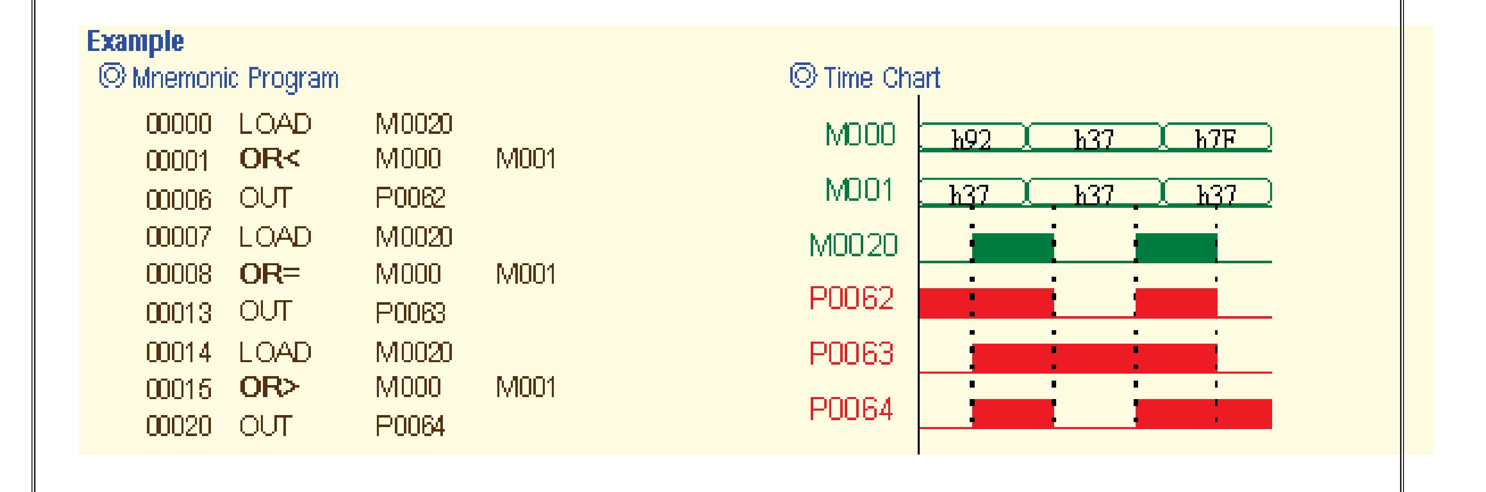
If the result of comparison between s١ and s٢ is satisfied with the condition, the result gets ON and performs AND operation with currently calculated result. If not, the result gets OFF and performs AND operation with currently calculated result.

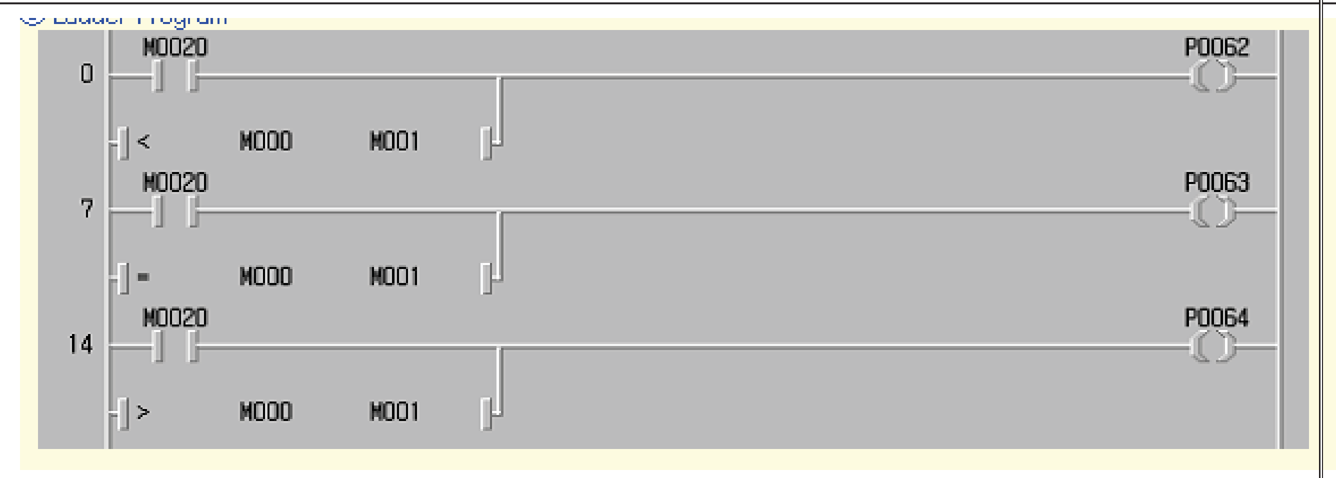




- **OR=** ٥ **OR Comparison Result (=)**
- **OR<>** ٥ **OR Comparison Result (<>)**
- **OR>=** ٥ **OR Comparison Result (>=)**
- **OR<=** ٥ **OR Comparison Result (<=)**
- **OR>** ٥ **OR Comparison Result (>)**
- **OR<** ٥ **OR Comparison Result (<)**
- **ORD=** ٥/٧/٩ **OR Comparison Result (=) (Double)**
- **ORD<>** ٥/٧/٩ **OR Comparison Result (<>) (Double)**
- **ORD>=** ٥/٧/٩ **OR Comparison Result (>=) (Double)**
- **ORD<=** ٥/٧/٩ **OR Comparison Result (<=) (Double)**
- **ORD>** ٥/٧/٩ **OR Comparison Result (>) (Double)**
- **ORD>** ٥/٧/٩ **OR Comparison Result (<) (Double)**

If the result of comparison between s١and s٢ is satisfied with the condition, the result gets ON and performs OR operation with currently calculated result. If not, the result gets OFF and performs OR operation with currently calculated result.

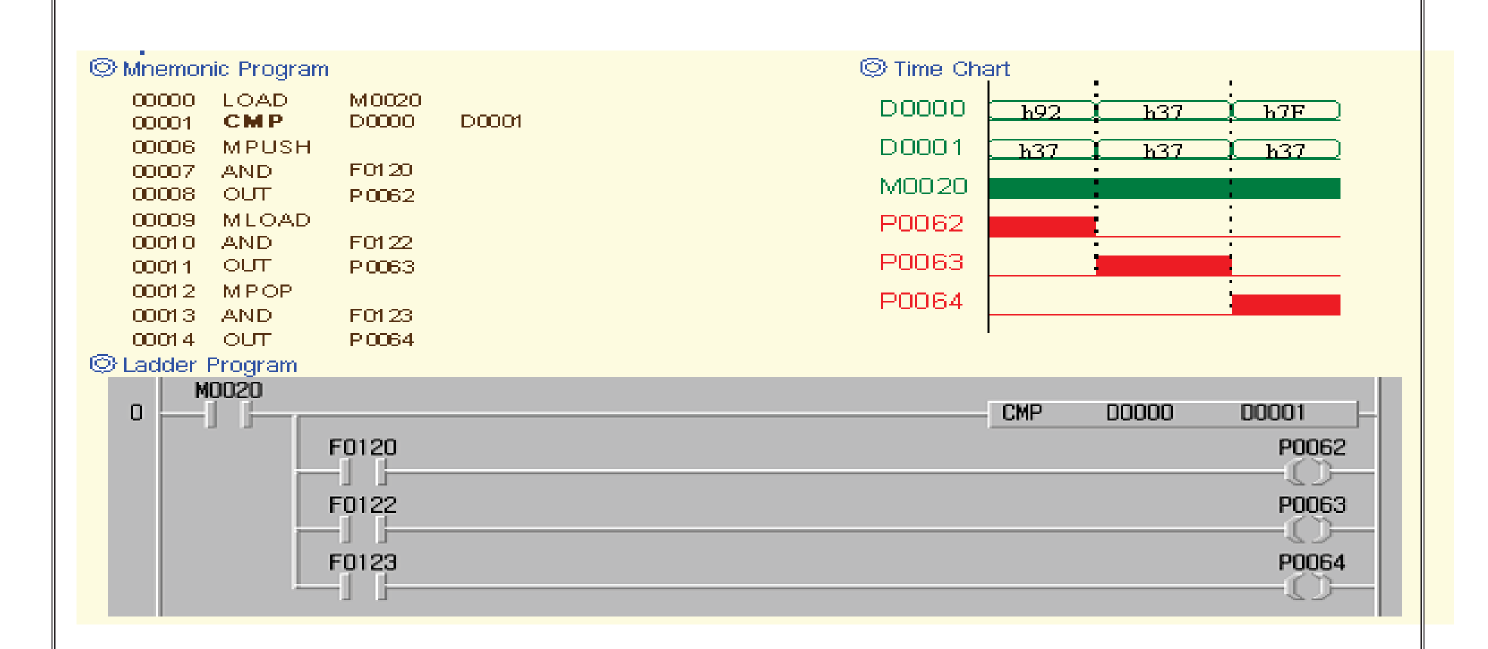




- **CMP** ٥ **Data Comparison**
- **CMPP** ٥ **Data Comparison (Pulse)**
- **DCMP** ٥/٧/٩ **Data Comparison (Double)**
- **DCMPP** ٥/٧/٩ **Data Comparison (Double & Pulse)**

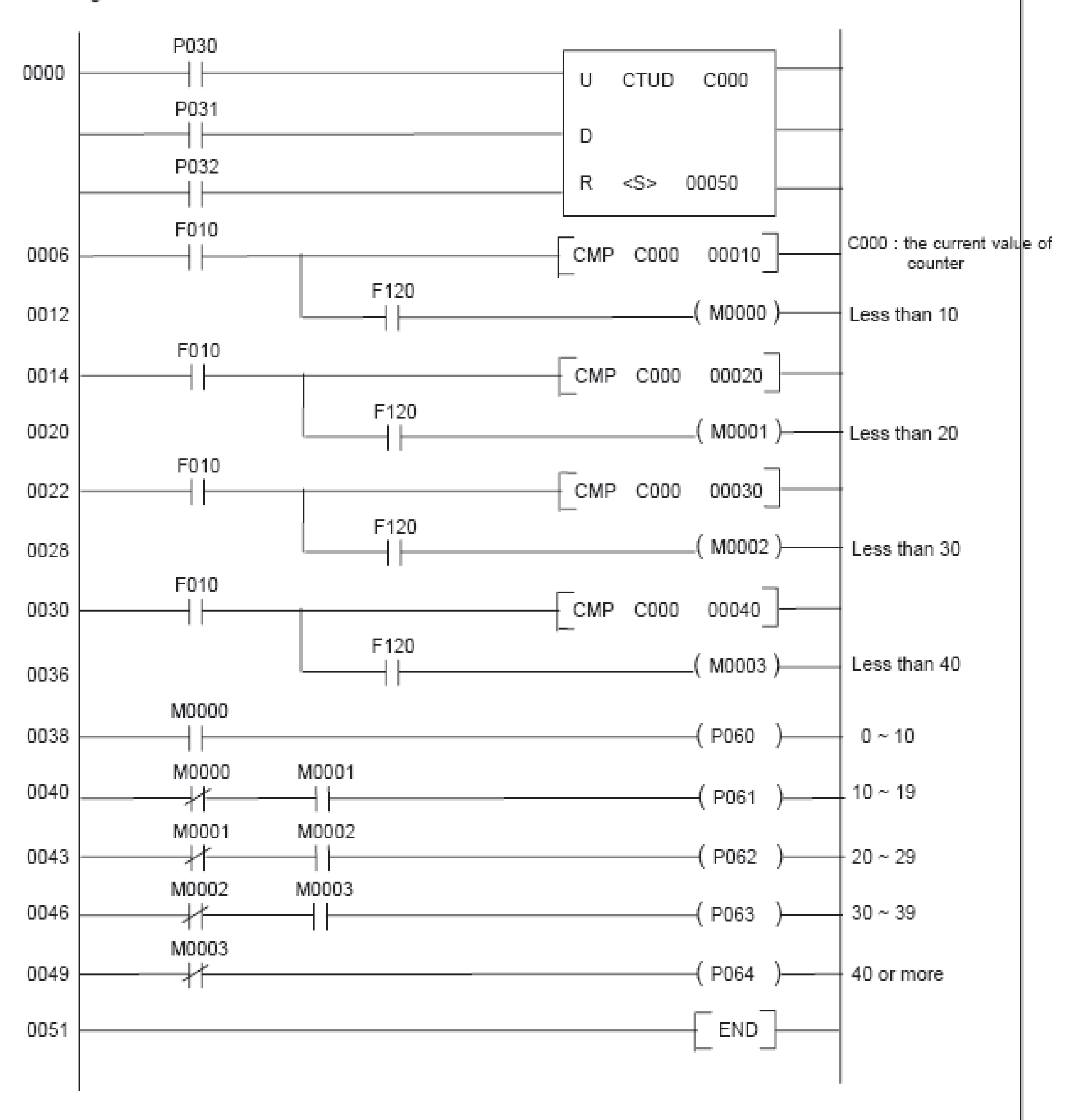
Compares the data of s١ and s٢ and sets the relevant flag among ٦ relays .

Flags	F120	F121	F122	F123	F124	F125
Operator	<	<=	=	>	>=	<>
s1>s2	0	0	0	1	1	1
s1<s2	1	1	0	0	0	1
s1=s2	0	1	1	0	1	0



Comparing circuit (example of CMP instruction)

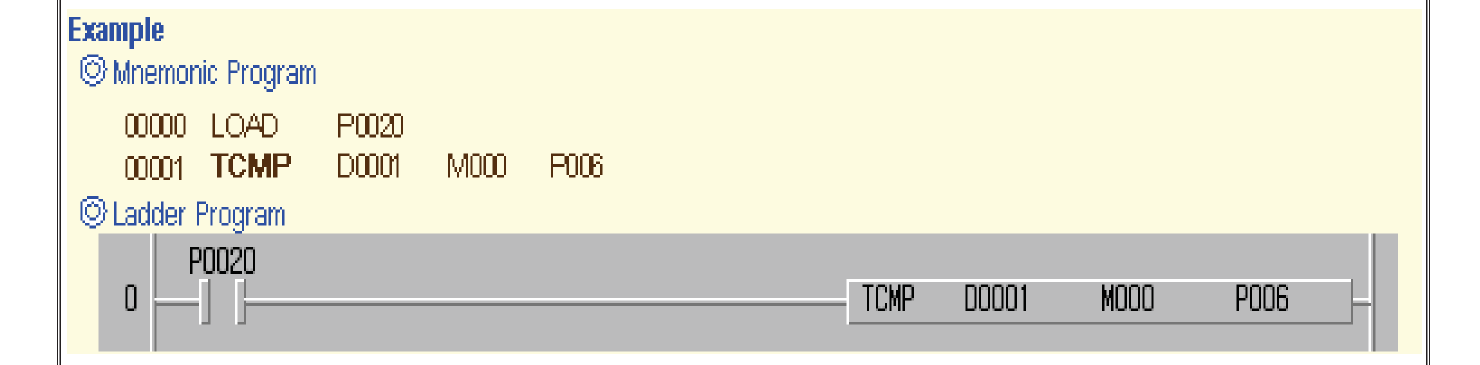
١. Operation There is a up-down counter C٠٠٠. P٠٣٠ is up-count input, and P٠٣١ is down-count input. If the current value of timer is ٠ ~ ٩, P٠٦٠ turns on. If the current value is ١٠ ~ ١٩, P٠٦١ will be on. P٠٦٢ will turn on when ٢٠ ~ ٢٩, P٠٦٣ will turn on when ٣٠ ~ ٣٩, and P٠٦٤ will be on when the current value is ٤٠ or larger.



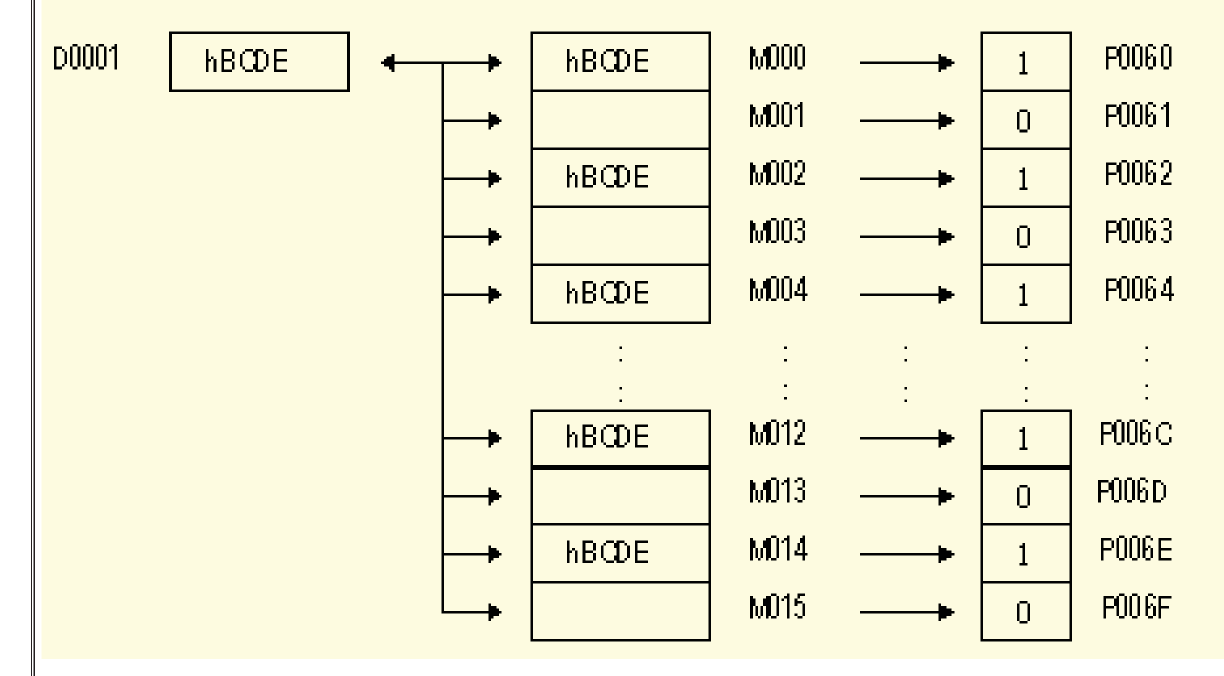
- **TCMP** ٥ **Table Comparison**
- **TCMPP** ٥ **Table Comparison (Pulse)**
- **DTCMP** ٧/٩ **Table Comparison (Double)**
- **DTCMPP** ٧/٩ **Table Comparison (Double & Pulse)**

١- Compares the data in the specified device s١ with each data of table (١٦ words) which starts with s٢ and the result is outputted to each bit (١٦bits) of the specified device d.

٢- If the result is identical, the bit data gets ' ١ ', if not, gets ' ٠ '.



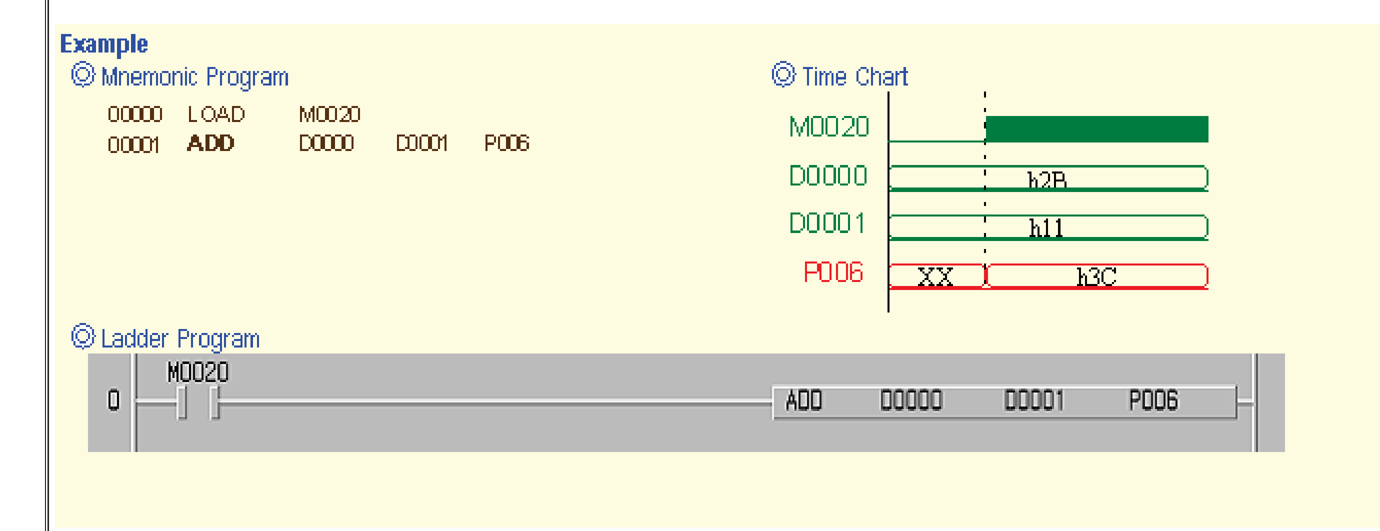
When P0020 gets ' ON ', compares data ' hBCDE ' in the device D0001 with each data of 16 words which starts with device M000 and the result is outputted to each bit (16bits) of the device P006.



Arithmetic Command

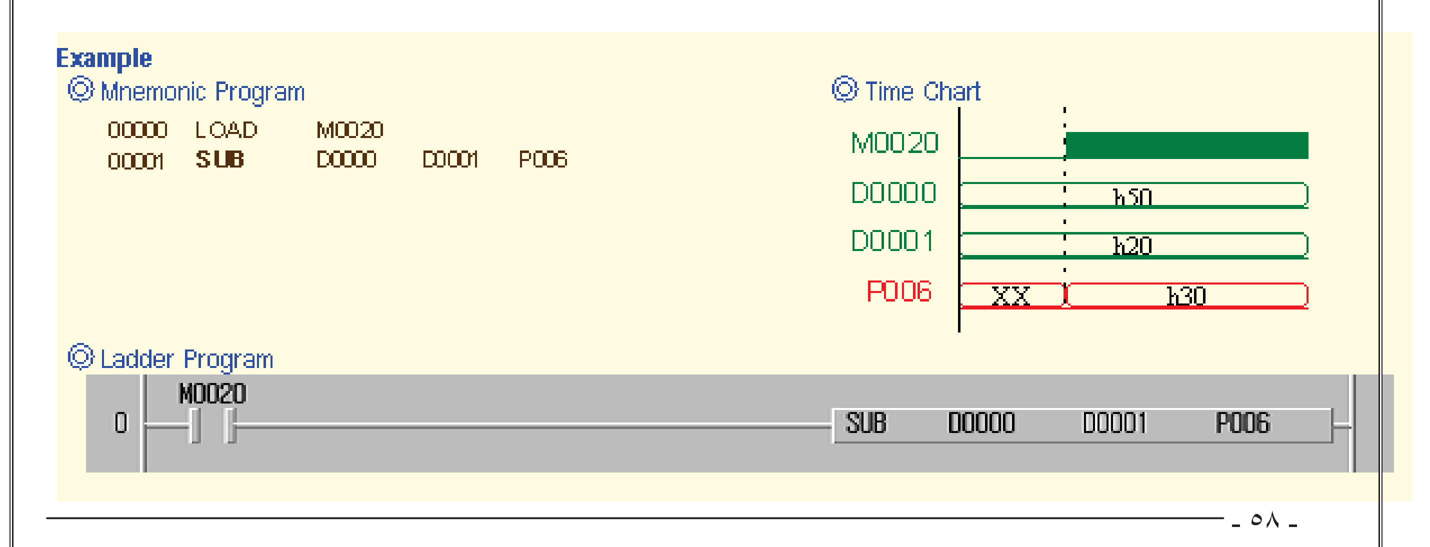
- **ADD** ٧ **Binary Addition**
- **ADDP** ٧ **Binary Addition (Pulse)**
- **DADD** ٧/٩/١١ **Binary Addition (Double)**
- **DADDP** ٧/٩/١١ **Binary Addition (Double & Pulse)**

Adds two words data specified as s١ and s٢ and stores the result in the specified device d.



- **SUB** ٧ **Binary Subtraction**
- **SUBP** ٧ **Binary Subtraction (Pulse)**
- **DSUB** ٧/٩/١١ **Binary Subtraction (Double)**
- **DSUBP** ٧/٩/١١ **Binary Subtraction (Double & Pulse)**

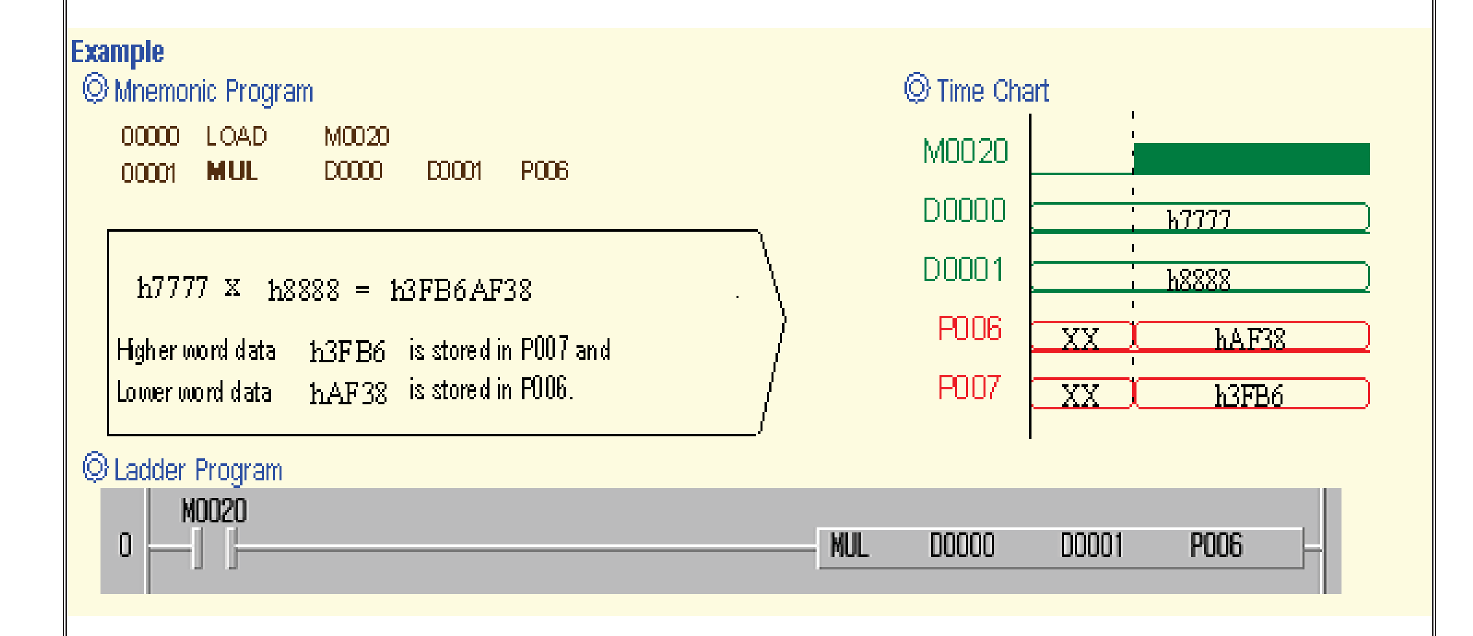
Subtracts the word data in the specified device s٢ from the word data in the specified device s١ and stores the result in the specified device d.



- **MUL** ٧ **Binary Multiplication**
- **MULP** ٧ **Binary Multiplication (Pulse)**
- **DMUL** ٧/٩/١١ **Binary Multiplication (Double)**
- **DMULP** ٧/٩/١١ **Binary Multiplication (Double & Pulse)**

١- Multiplies two word data specified as s١ and s٢ and stores the result in the specified device d and d+١.

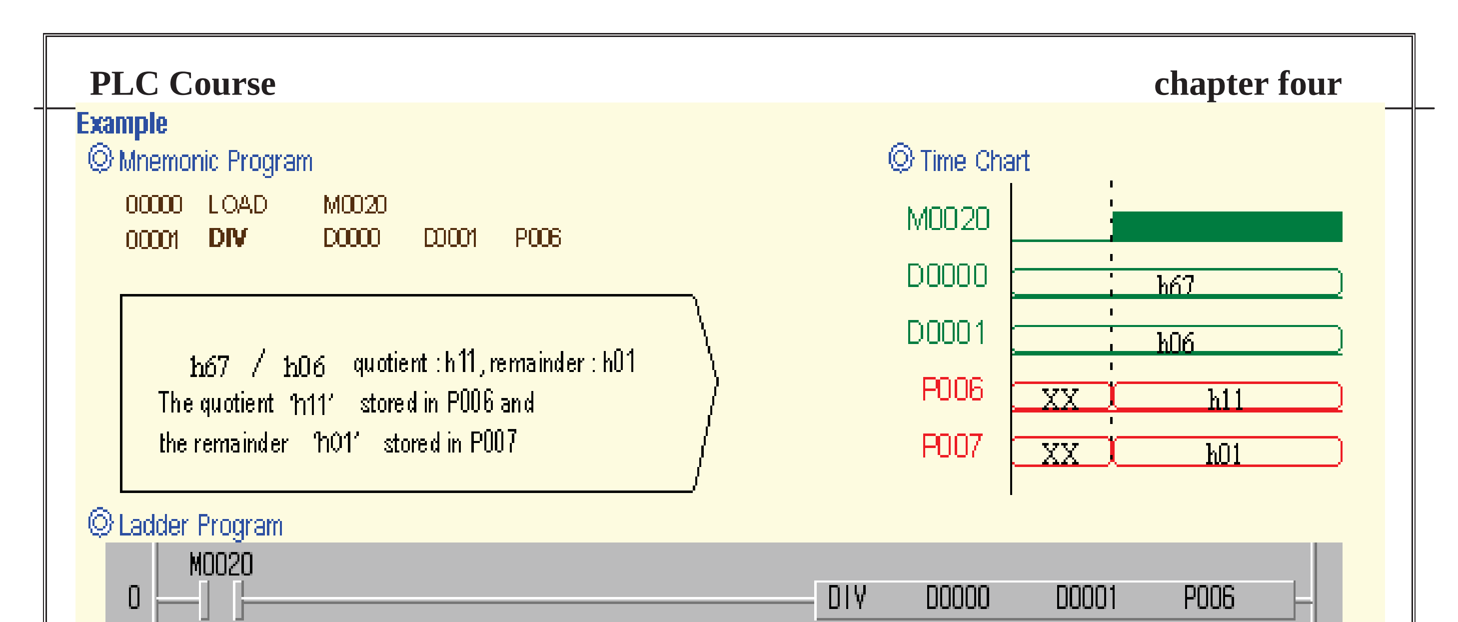
٢- The lower ٤ digit data of the result is stored in the device d and the higher ٤ digit data is stored in the device d+١.



- **DIV** ٧ **Binary Division**
- **DIVP** ٧ **Binary Division (Pulse)**
- **DDIV** ٧/٩/١١ **Binary Division (Double)**
- **DDIVP** ٧/٩/١١ **Binary Division (Double & Pulse)**

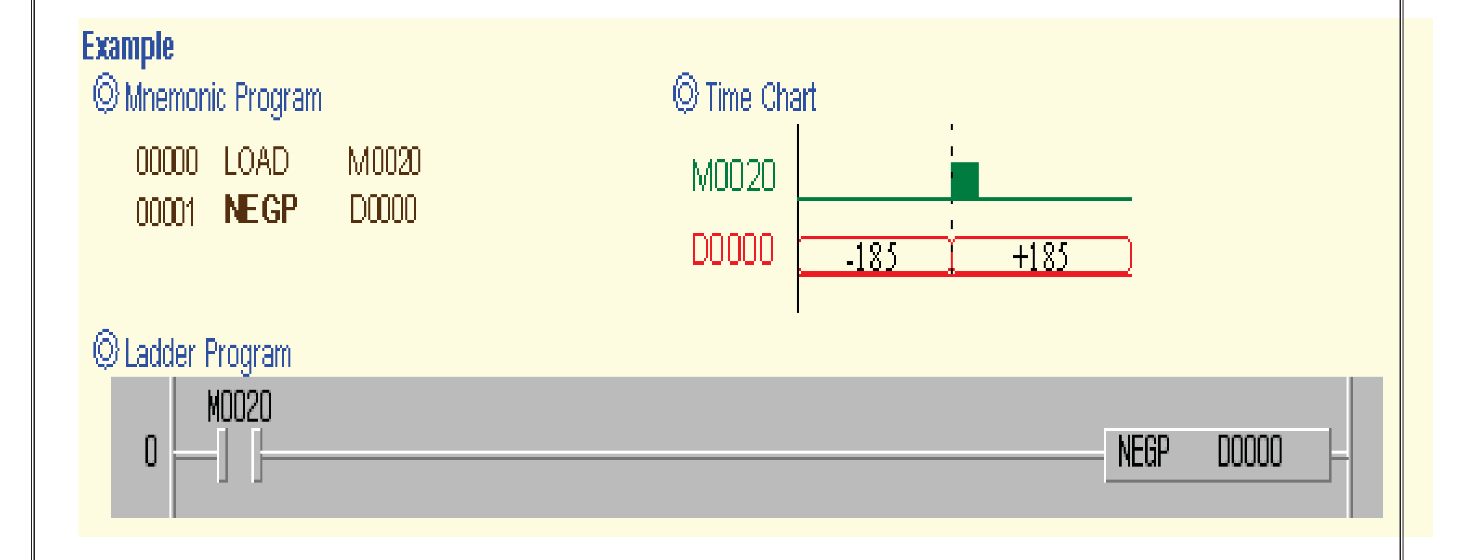
١-Divides the word data in the specified device s١ by the word data in the specified device s٢ and stores the result in the specified device d and d+١.

٢-The quotient is stored in the device d and the remainder is stored in the device d+١.



- **NEG ٣ Negative**
- **NEGP ٣ Negative (Pulse)**
- **DNEG ٣ Negative (Double)**
- **DNEGP ٣ Negative (Double & Pulse)**

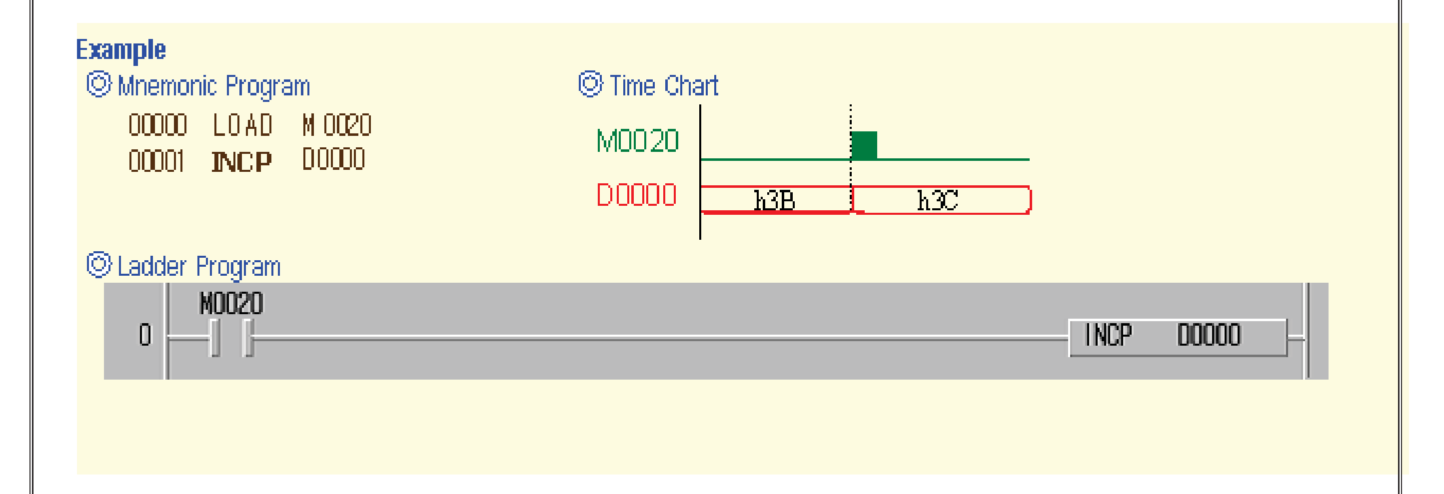
Changes the sign of the data in the specified device d, i.e. changes ‘ + ’ to ‘ - ‘ and ‘–‘ to ‘ + ’.



Increment/Decrement Command

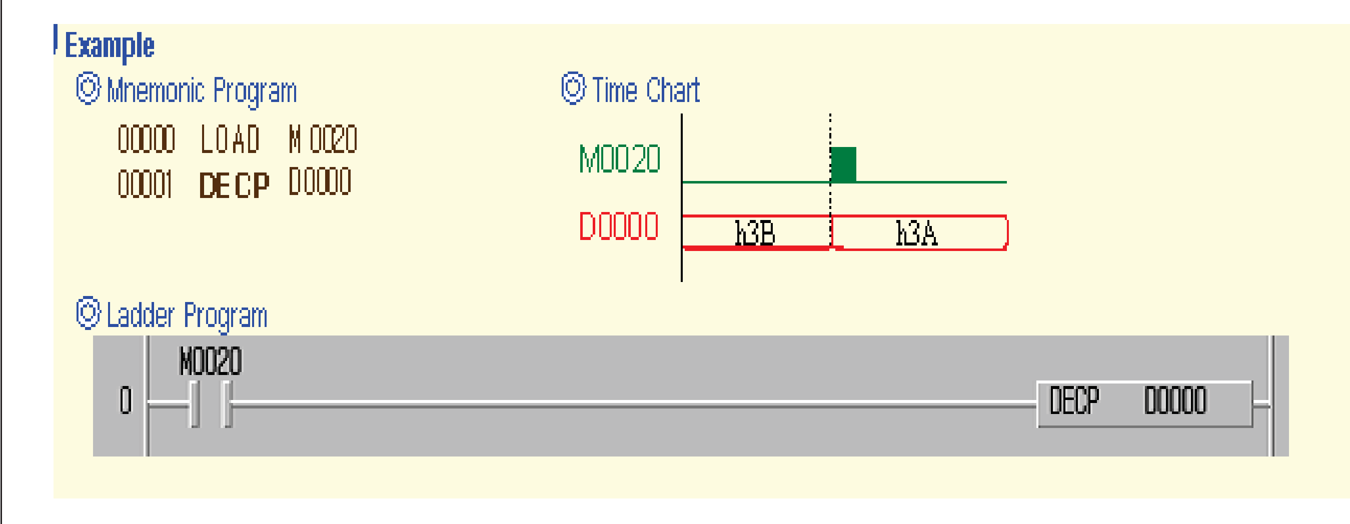
- **INC** ٣ **Increment by one**
- **INCP** ٣ **Increment by one (Pulse)**
- **DINC** ٣ **Increment by one (Double)**
- **DINCP** ٣ **Increment by one (Double & Pulse)**

Adds ‘ ١ ’ to the data of the specified device d and stores the result in the device d.



- **DEC** ٣ **Decrement by one**
- **DECP** ٣ **Decrement by one (Pulse)**
- **DDEC** ٣ **Decrement by one (Double)**
- **DDECP** ٣ **Decrement by one (Double & Pulse)**

Subtracts ‘ ١ ’ to the data of the specified device d and stores the result in the device d.



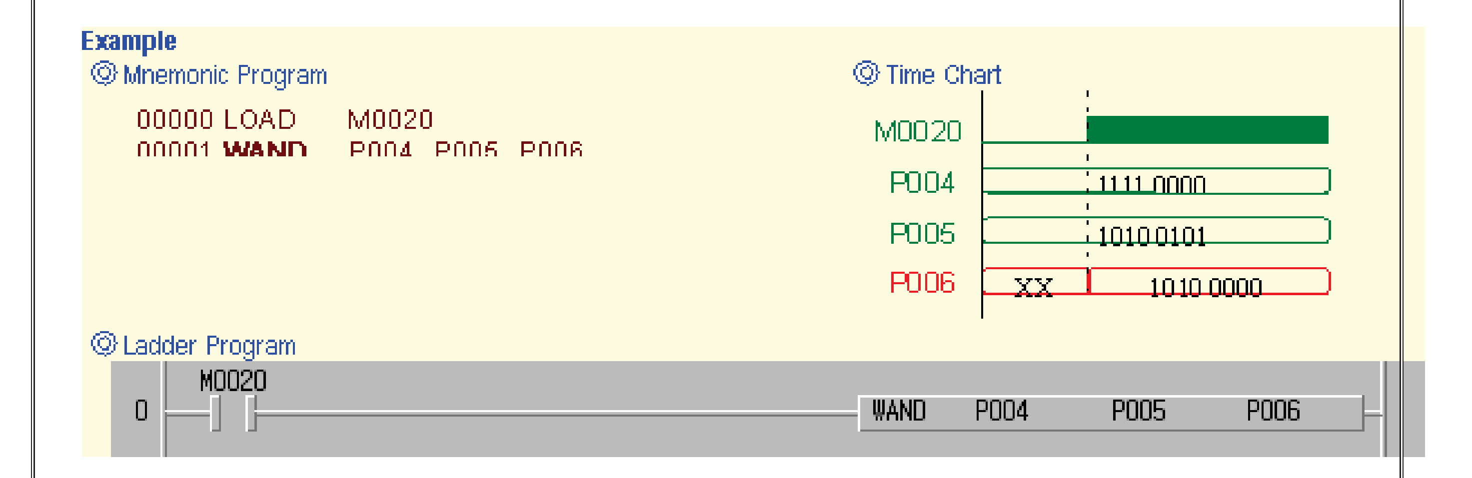
Logical Operation Command

- **WAND** ٧ **Word Data AND Operation**
- **WANDP** ٧ **Word Data AND Operation (Pulse)**
- **DWAND** ٧/٩/١١ **Word Data AND Operation (Double)**
- **DWANDP** ٧/٩/١١ **Word Data AND Operation (Double & Pulse)**

After executing AND operation for each bit data of s١ and s٢, the result is saved in the device d for each bit data.

This table shows the result of AND operation for each bit data.

bit data of s1	1	1	0	0
bit data of s2	1	0	1	0
Operation Result	1	0	0	0

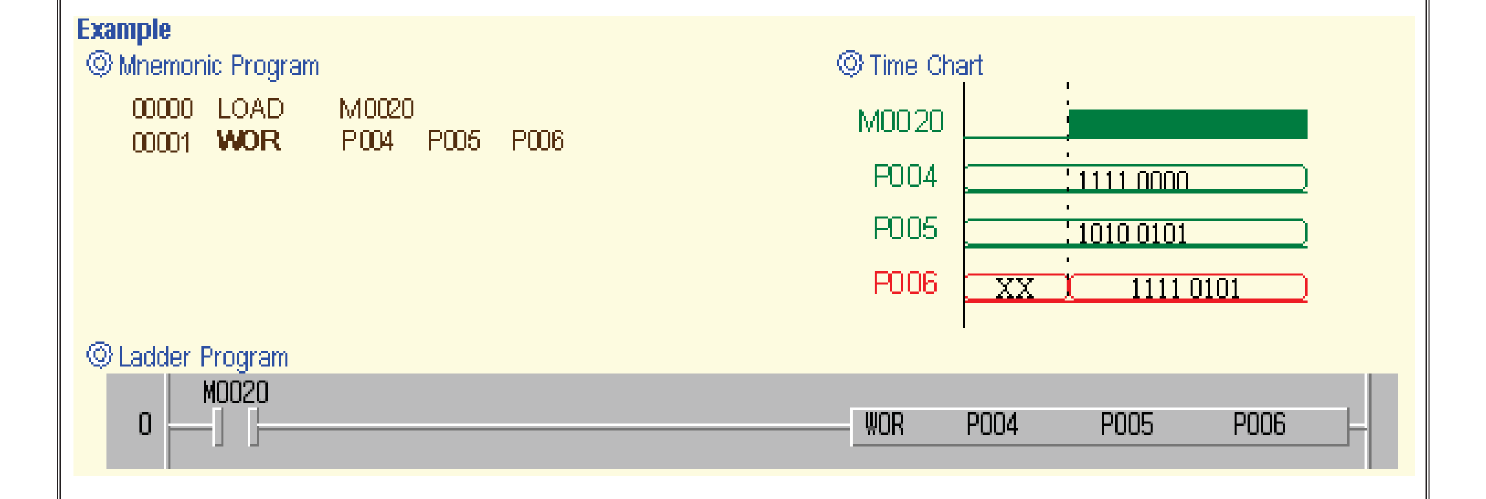


- **WOR** ٧ **Word Data OR Operation**
- **WORP** ٧ **Word Data OR Operation (Pulse)**
- **DWOR** ٧/٩/١١ **Word Data OR Operation (Double)**
- **DWORP** ٧/٩/١١ **Word Data OR Operation (Double & Pulse)**

After executing OR operation for each bit data of device s١ and s٢, the result is saved in the device d for each bit data.

This table shows the result of OR operation for each bit data.

bit data of s1	1	1	0	0
bit data of s2	1	0	1	0
Operation Result	1	1	1	0



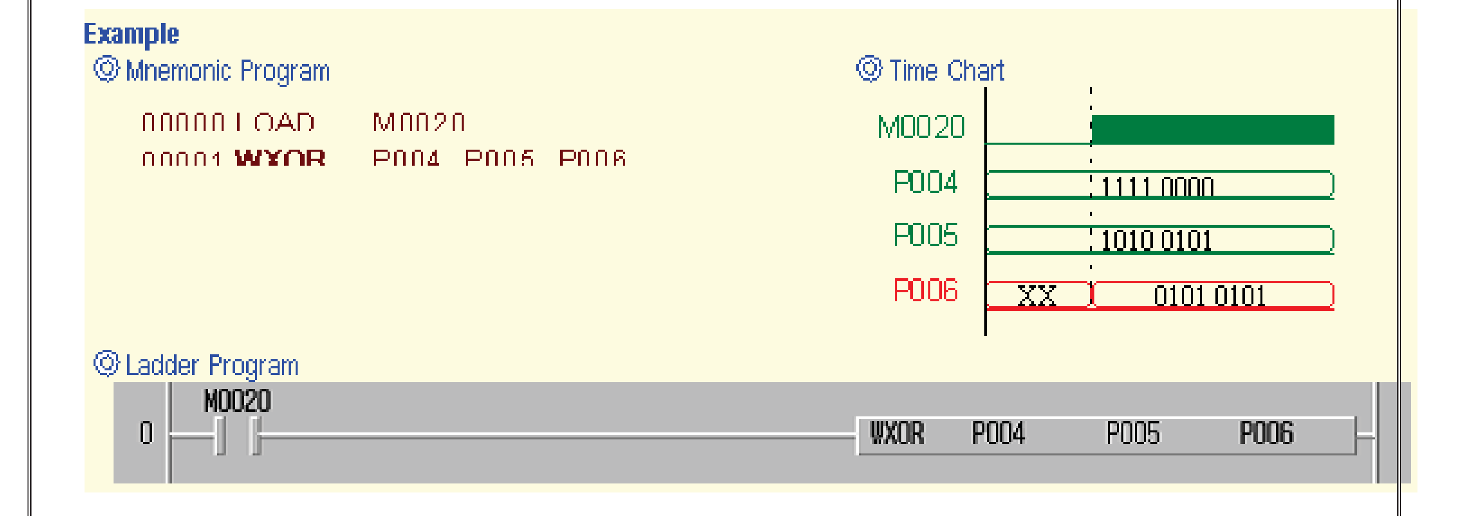
- **WXOR** ٧ **Word Data Exclusive OR Operation**
- **WXORP** ٧ **Word Data Exclusive OR Operation (Pulse)**
- **DWXOR** ٧/٩/١١ **Word Data Exclusive OR Operation (Double)**
- **DWXORP** ٧/٩/١١ **Word Data Exclusive OR Operation (Double & Pulse)**

After executing XOR operation for each bit data of s١ and s٢, the result is saved in the device d for each bit data.

This table shows the result of XOR operation for each bit data.

Bit data of S١	١	١	٠	٠
Bit data of S٢	١	٠	١	٠
Operation result	٠	١	١	٠

XOR : Exclusive OR



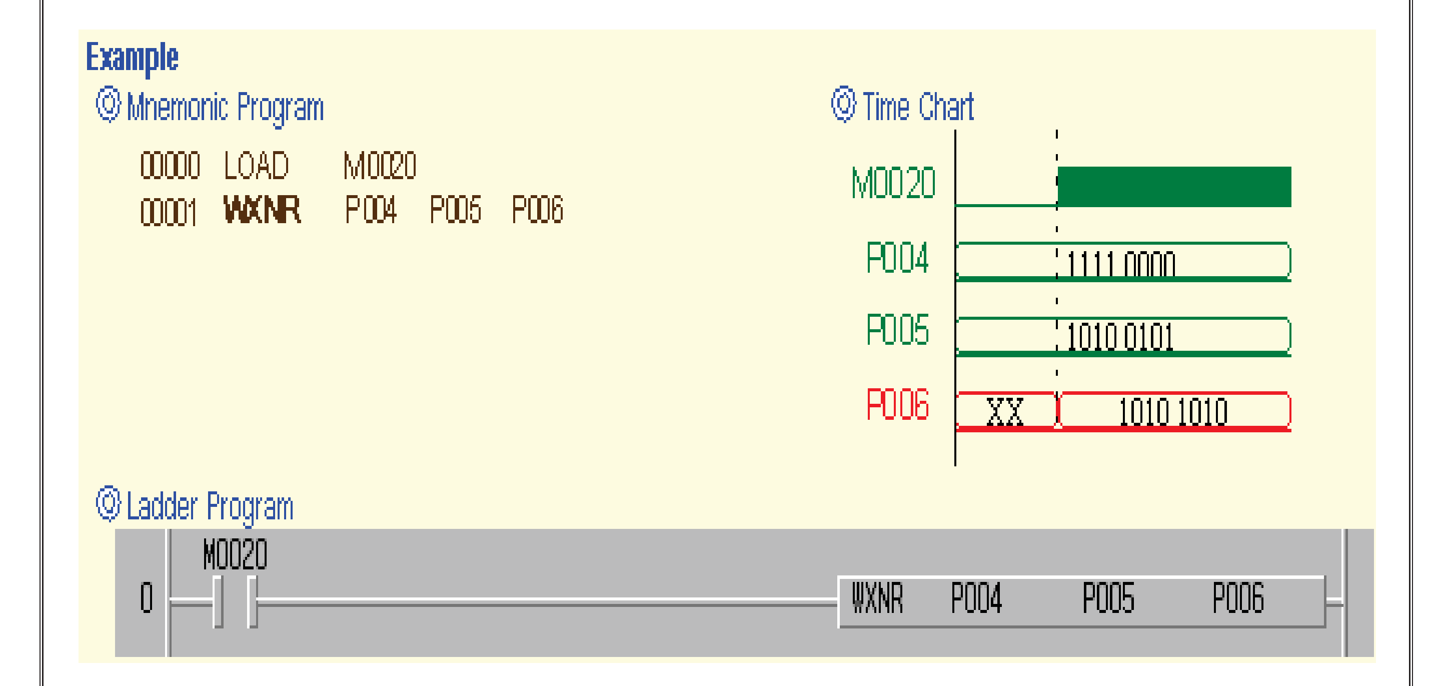
- **WXNR** ٧ **Word Data Exclusive NOR Operation**
- **WXNRP** ٧ **Word Data Exclusive NOR Operation (Pulse)**
- **DWXNR** ٧/٩/١١ **Word Data Exclusive NOR Operation (Double)**
- **DWXNRP** ٧/٩/١١ **Word Data Exclusive NOR Operation (Double & Pulse)**

After executing XNOR operation for each bit data of s١ and s٢, the result is saved in the device d for each bit data.

This table shows the result of XNOR operation for each bit data.

Bit data of S١	١	١	٠	٠
Bit data of S٢	١	٠	١	٠
Operation result	١	٠	٠	١

XNOR : Exclusive NOR

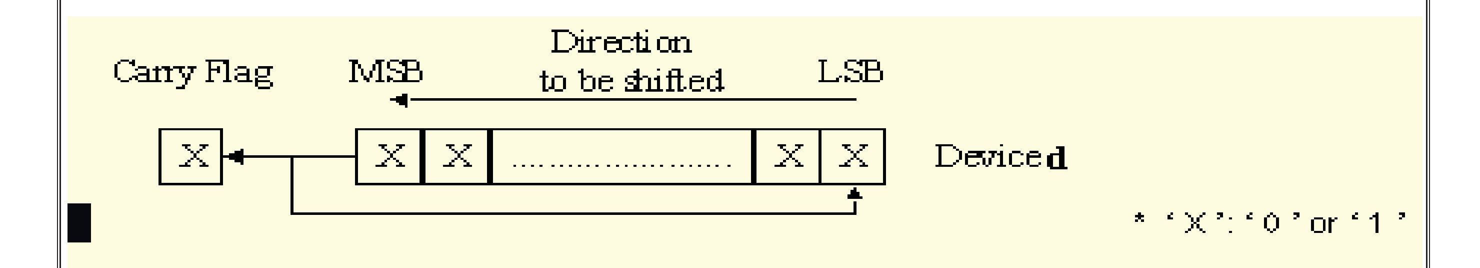


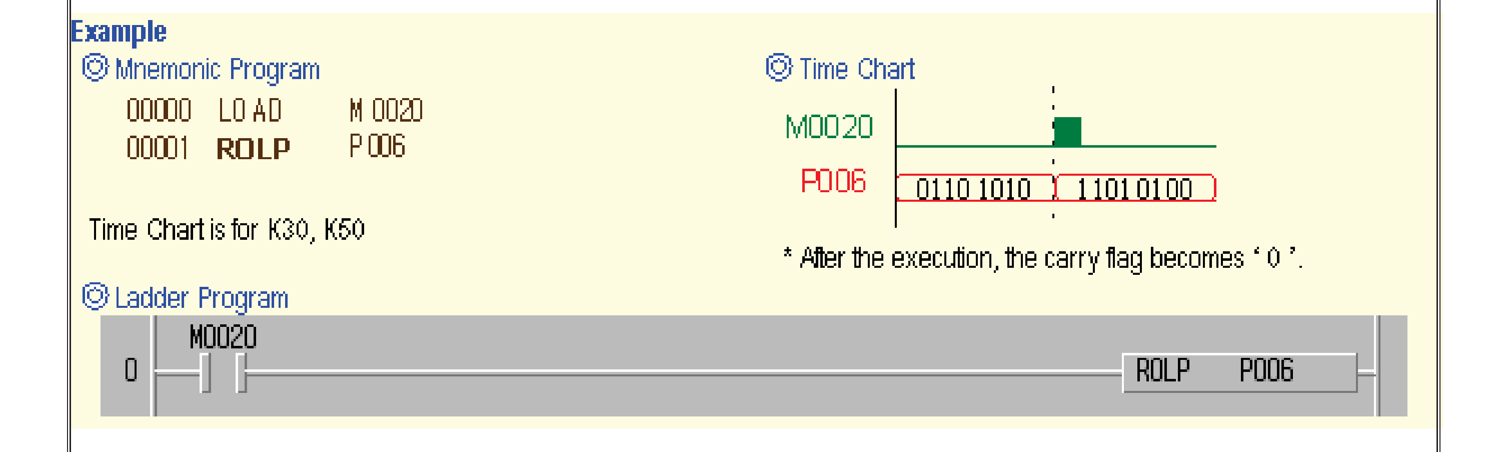
Rotate/Shift Command

- **ROL** ٣ **Rotate Left**
- **ROLP** ٣ **Rotate Left (Pulse)**
- **DROL** ٣ **Rotate Left (Double)**
- **DROLP** ٣ **Rotate Left (Double & Pulse)**

١. Rotates the bit data in the specified device d to the left by one bit.

٢. The data of the last output bit(MSB) is stored in the carry flag and the LSB as below.

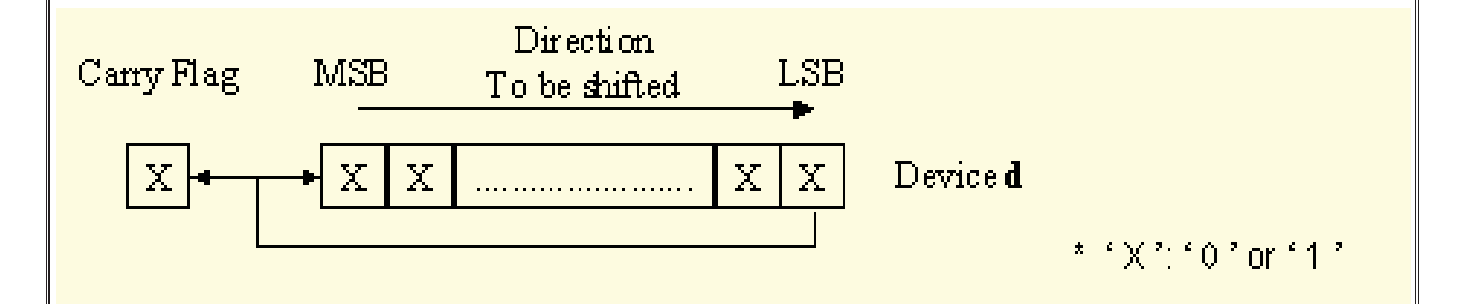


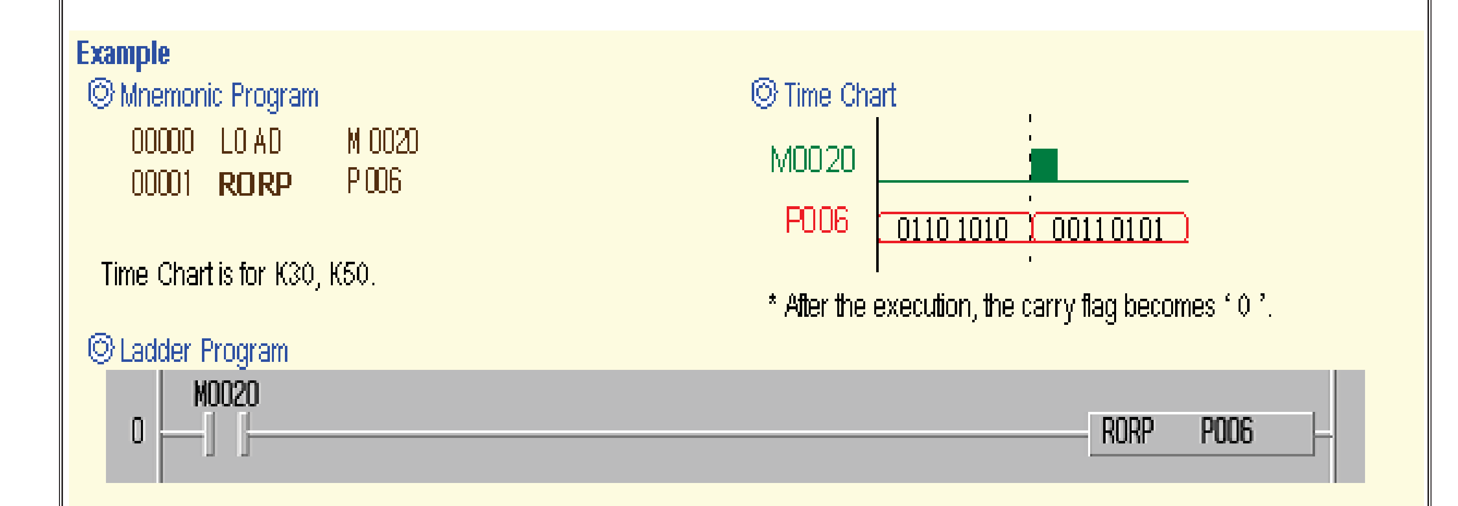


- **ROR** ٣ **Rotate Right**
- **RORP** ٣ **Rotate Right (Pulse)**
- **DROR** ٣ **Rotate Right(Double)**
- **DRORP** ٣ **Rotate Right (Double & Pulse)**

١. Rotates the bit data in the specified device d to the right by one bit.

٢. The data of the LSB is stored in the carry flag and the MSB as below .

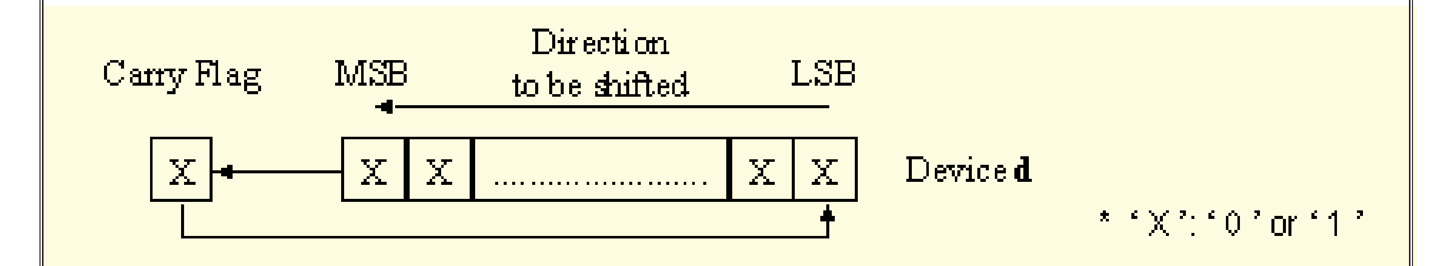


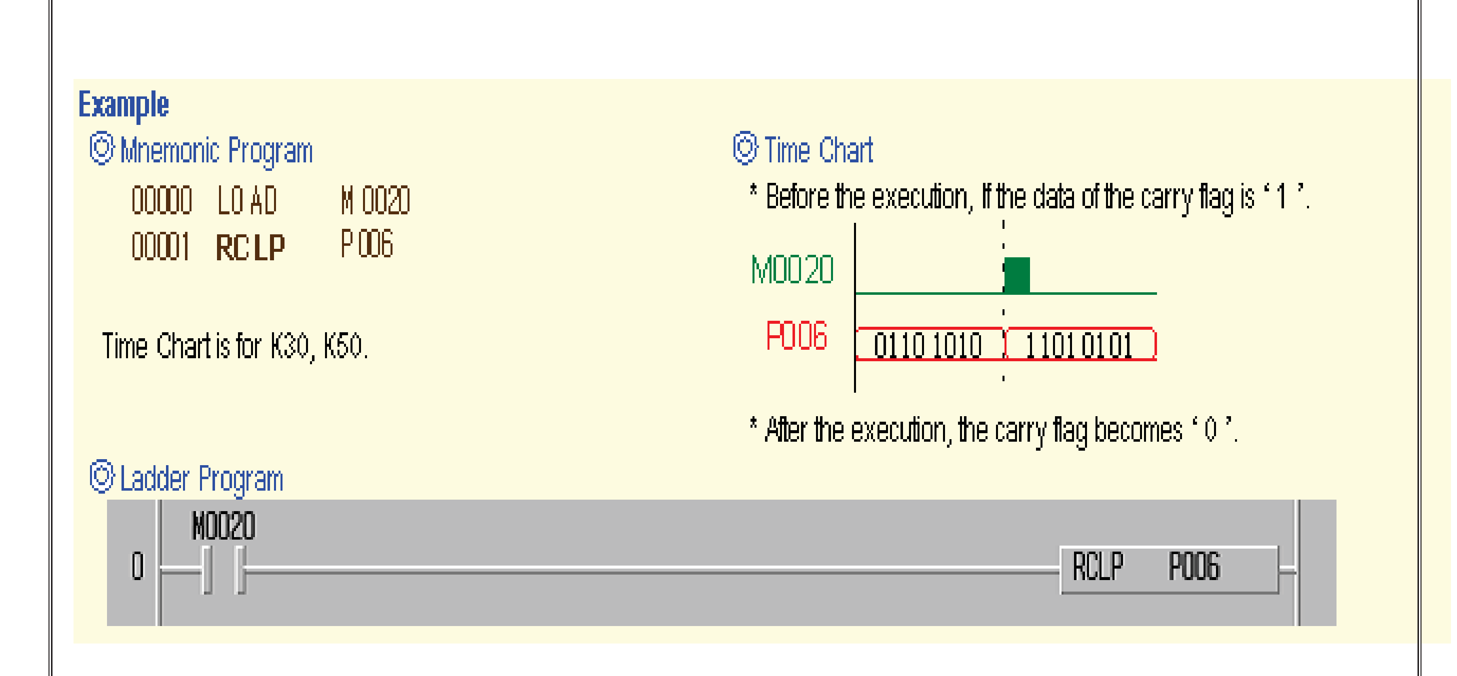


- **RCL** ٣ **Rotate Left with Carry**
- **RCLP** ٣ **Rotate Left with Carry (Pulse)**
- **DRCL** ٣ **Rotate Left with Carry (Double)**
- **DRCLP** ٣ **Rotate Left with Carry (Double & Pulse)**

١. Rotates the bit data in the specified device d to the left by one bit.

٢. The data of the last output bit(MSB) is stored in the carry flag and the data of the carry flag is shifted to the LSB as below .

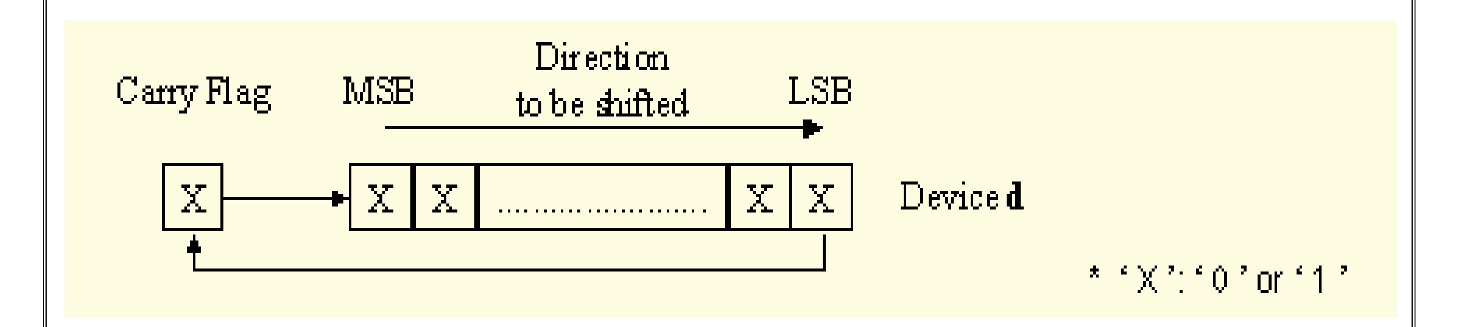


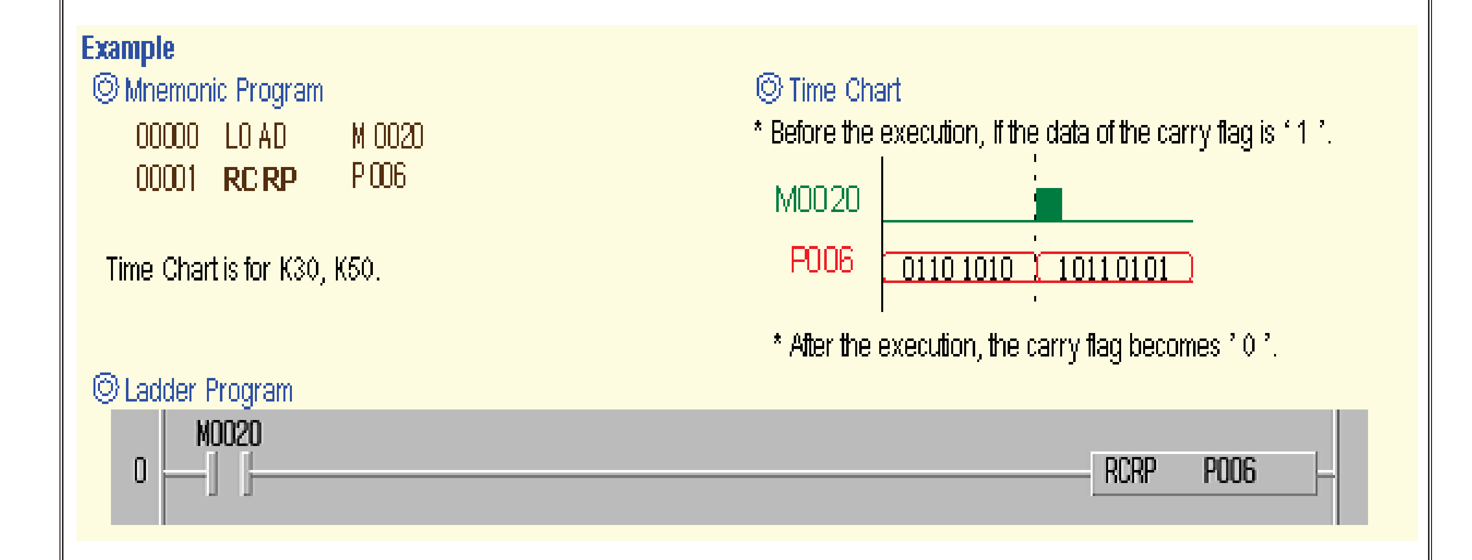


- **RCR** ٣ **Rotate Right with Carry**
- **RCRP** ٣ **Rotate Right with Carry (Pulse)**
- **DRCR** ٣ **Rotate Right with Carry (Double)**
- **DRCRP** ٣ **Rotate Right with Carry (Double & Pulse)**

١. Rotates the bit data in the specified device d to the right by one bit.

٢. The data of the LSB is stored in the carry flag and the carry flag is shifted to the LSB as below.



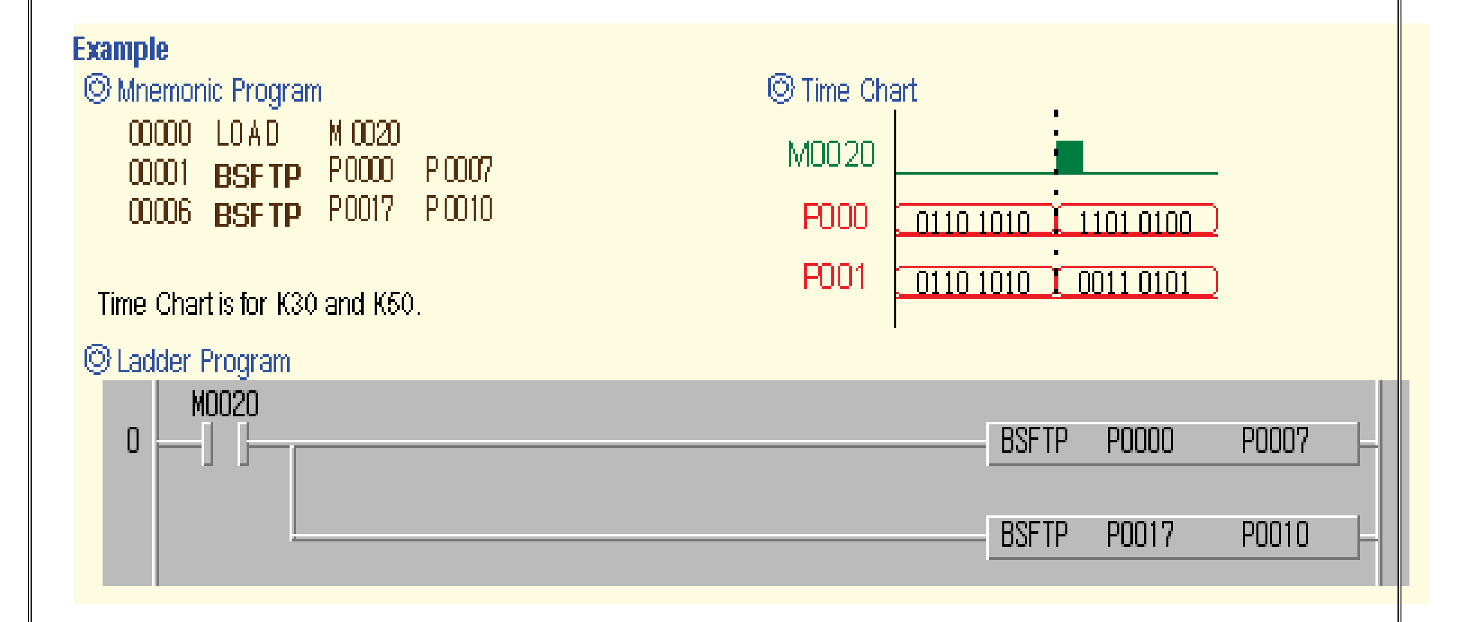


- **BSFT** ٥ **Bit Shift**
- **BSFTP** ٥ **Bit Shift (Pulse)**

١. Shifts a bit one by one from the start device s to the end device e.

٢. The direction of bit shift is determined by the order of the specified device. If s is greater than e, the bit shift is to the right and if s is less than e, the bit shift is to the left.

٣. After the shift, the start device is filled with ‘ ٠ ’.

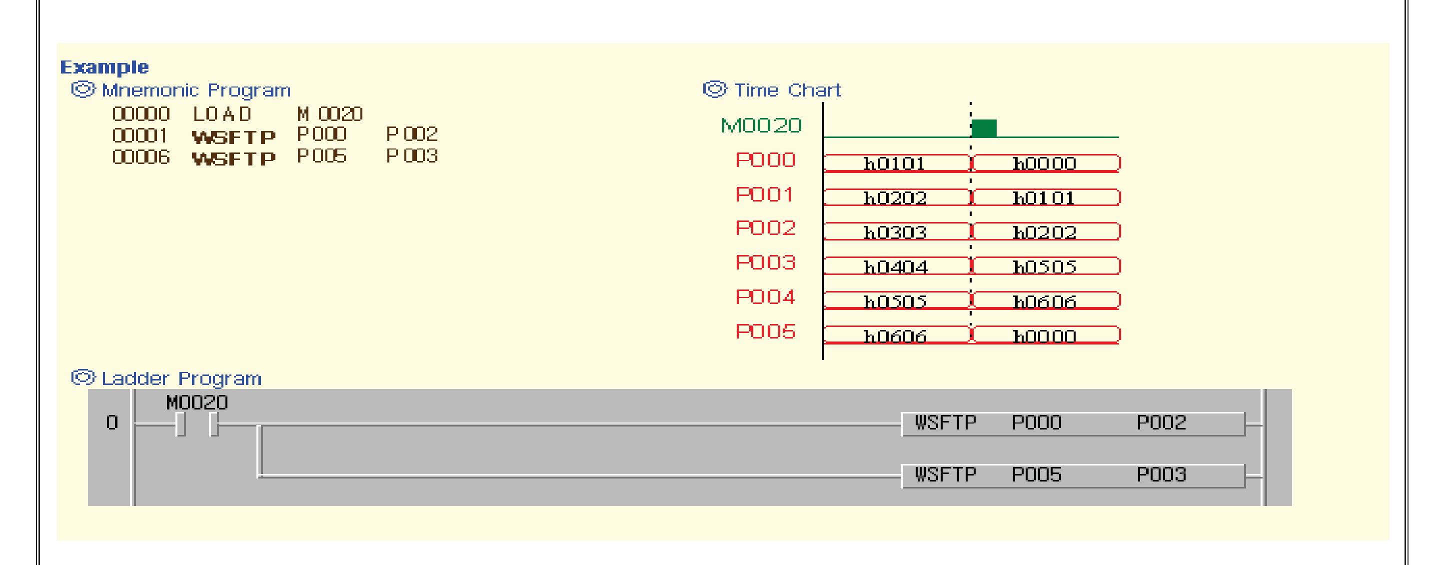


- **WSFT** ٥ **Word Shift**
- **WSFTP** ٥ **Word Shift (Pulse)**

١. Shift the word data one by one from the start device s to the end device e.

٢. The direction of data shift is determined by the order of the specified device. If s is greater than e, the bit shift is to the right and if s is less than e, the bit shift is to the left.

٣. After the data shift, the start device is filled with ‘ h٠٠٠٠ ’.



SR ٥ Shift Register

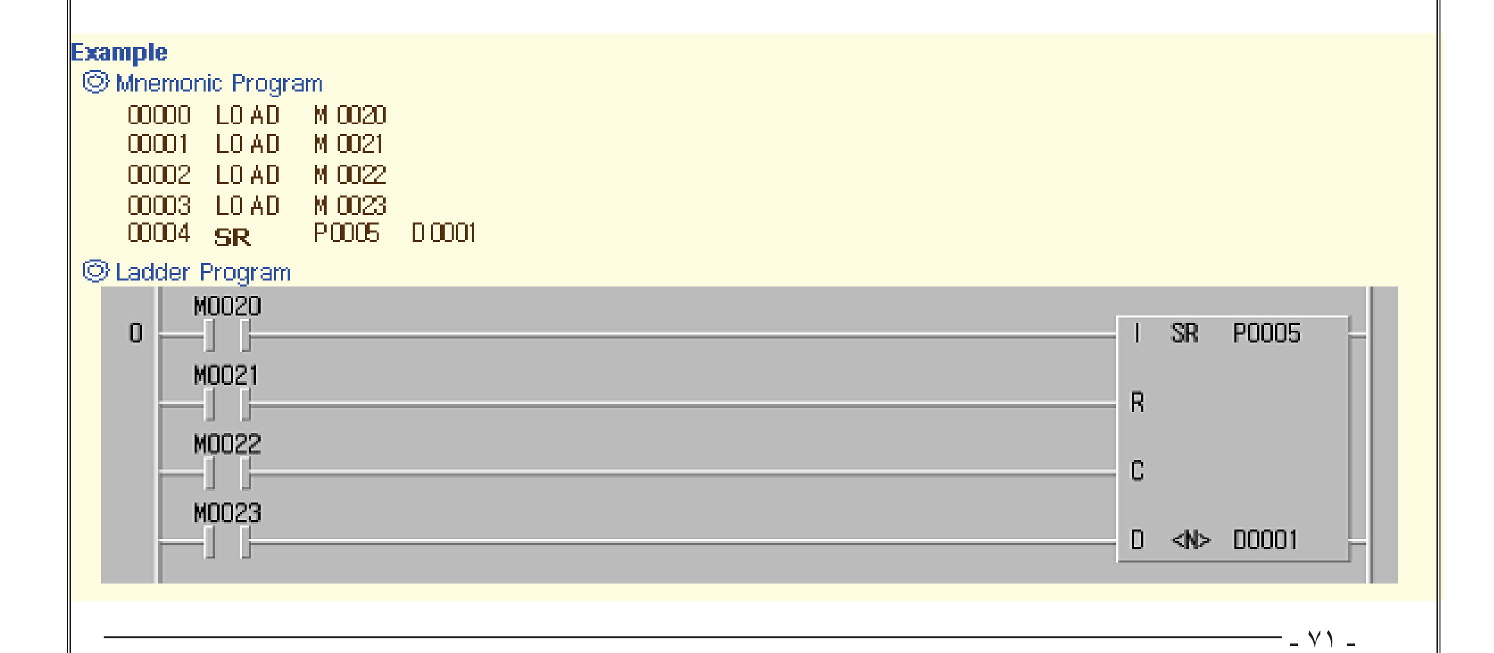
١. Shifts the bit data one by one as much as n from the start device d.

٢. The shift is occurred when the C(Clock) gets ON in the rising edge

٣. The shifting direction is determined by the input condition D(Direction), that is, When the input condition C goes from low to high, If D gets ON, the direction is to the right and if D gets OFF, the direction is to the left.

٤. After shifting the area, the blanked bit(most left or right) is filled with the input condition 'I(Input data)'.

٥. When R(Reset) gets ON, All devices as much as n from the start device d are cleared.



Jump/Interrupt Command

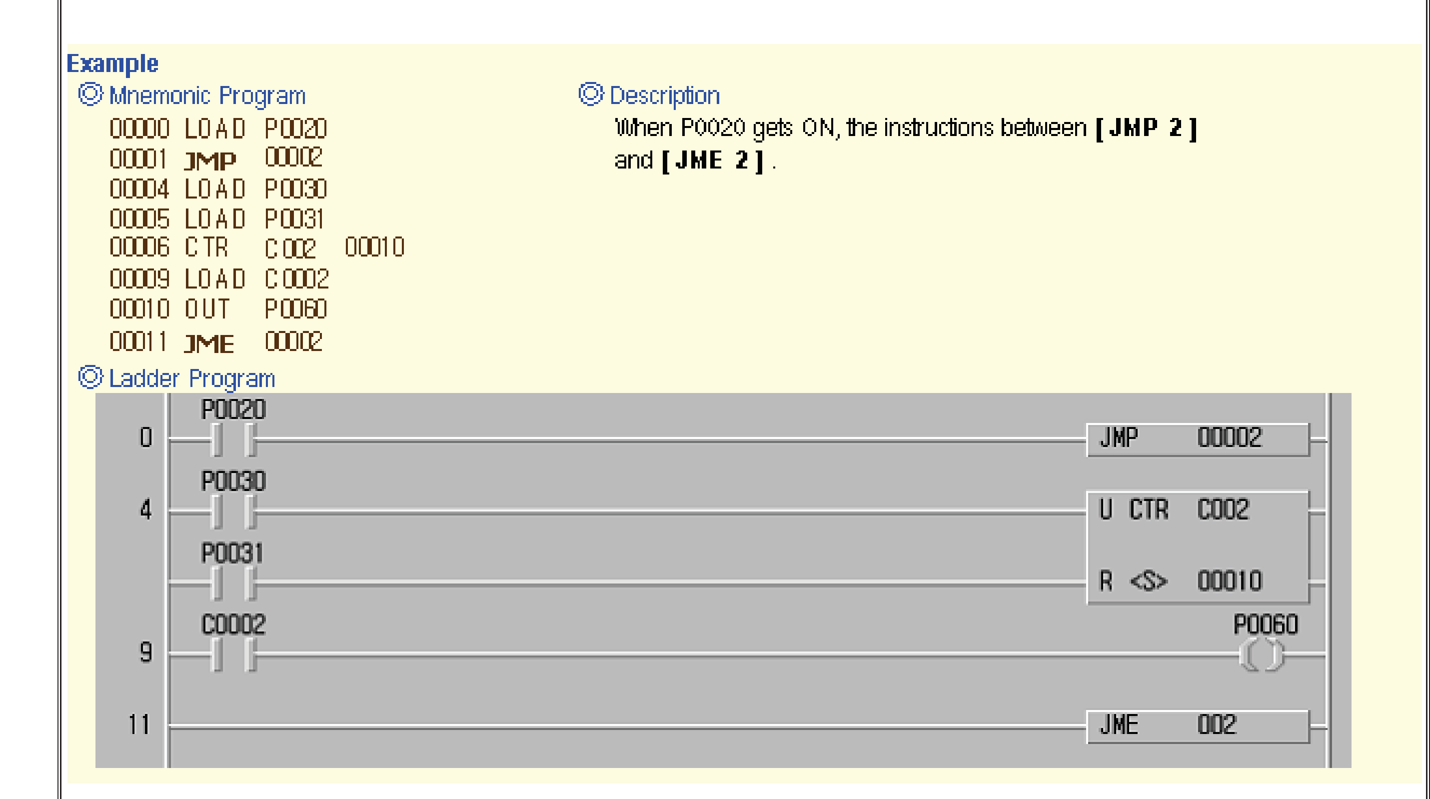
- **JMP ١/٣ Jump**
- **JME ١ End of Jump**

١. If the input condition of [JMP n] gets ON, the operation jumps to [JME n] and all instructions between [JMP n] and [JME n] are not executed.

٢. [JMP n] is matched with [JME n] which the number n is same each other.

٣. This command is applicable for the emergency.

٤. [JMP ٠] can be used repeatedly.



- **FOR** ٣ **Start of FOR ~ NEXT Loop**
- **NEXT** ١ **End of FOR ~ NEXT Loop**
- **BREAK** ٣ **Break of FOR ~ NEXT Loop**

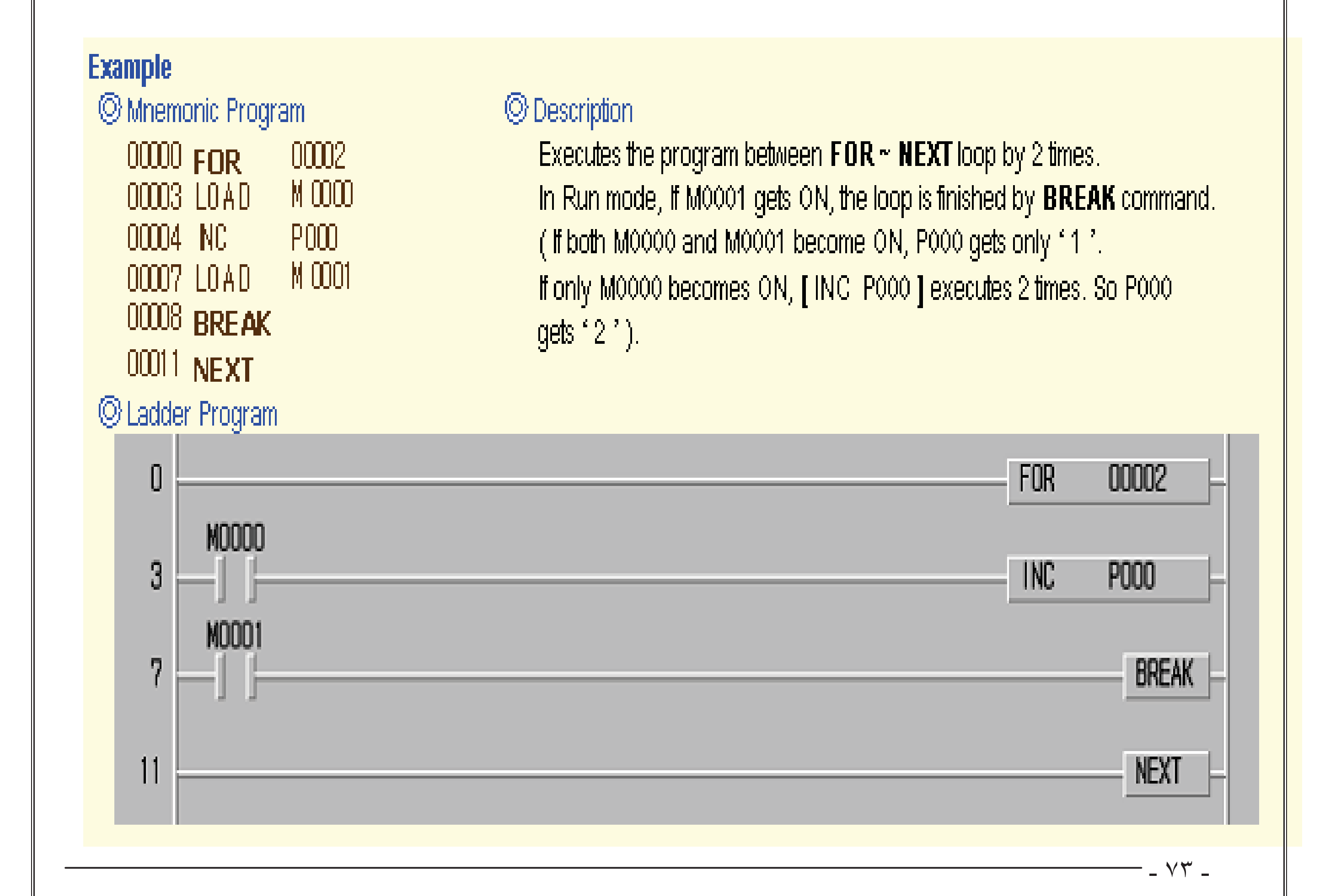
١. In Run mode, [FOR n] Command executes the program between [FOR n] and [NEXT] as much as n, the next step of [NEXT] command is executed.

٢. [FOR ~ NEXT] loop can be used and nested up to ٥ times within the same program.

٣. When END or RET Command is executed between [FOR ~ NEXT] loop, or [NEXT] Command is executed before [FOR n] Command, the loop is not executed.

٤. JMP n ~ JME n] Command is executed within the same [FOR ~ NEXT] loop and [FOR ~ NEXT] within [JMP n ~ JME n] is ignored.

٥. to finish [FOR ~ NEXT] loop, use [BREAK] Command.]



CALL	١/٣	Call Subroutine
CALLP	١/٣	Call Subroutine (Pulse)
SBRT	١/٣	Start from Subroutine
RET	١/٣	Return from Subroutine

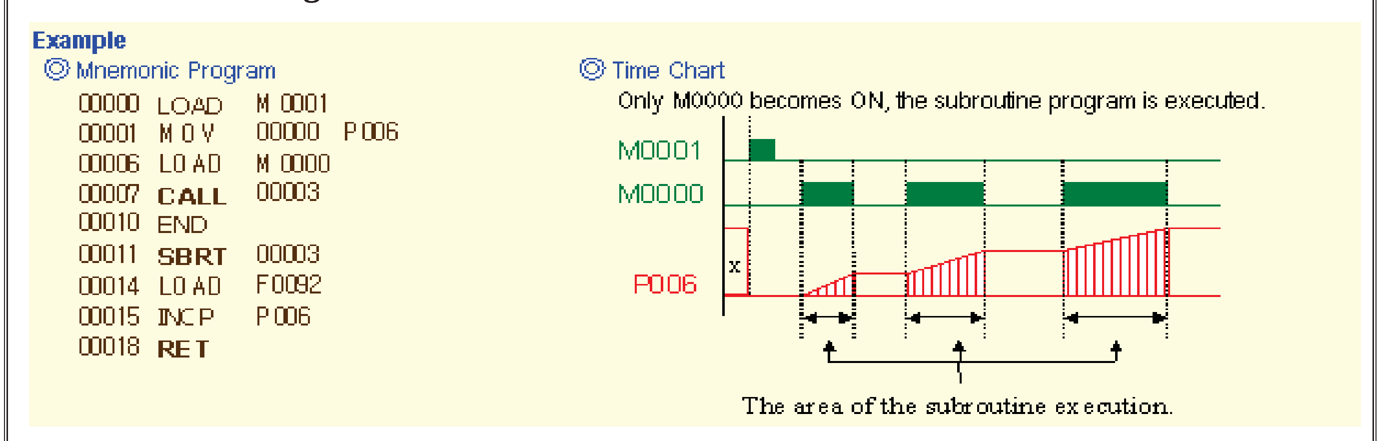
١. The subroutine program between [SBRT n] and [RET] command is executed by calling of [CALL n] or [CALLP n] command.

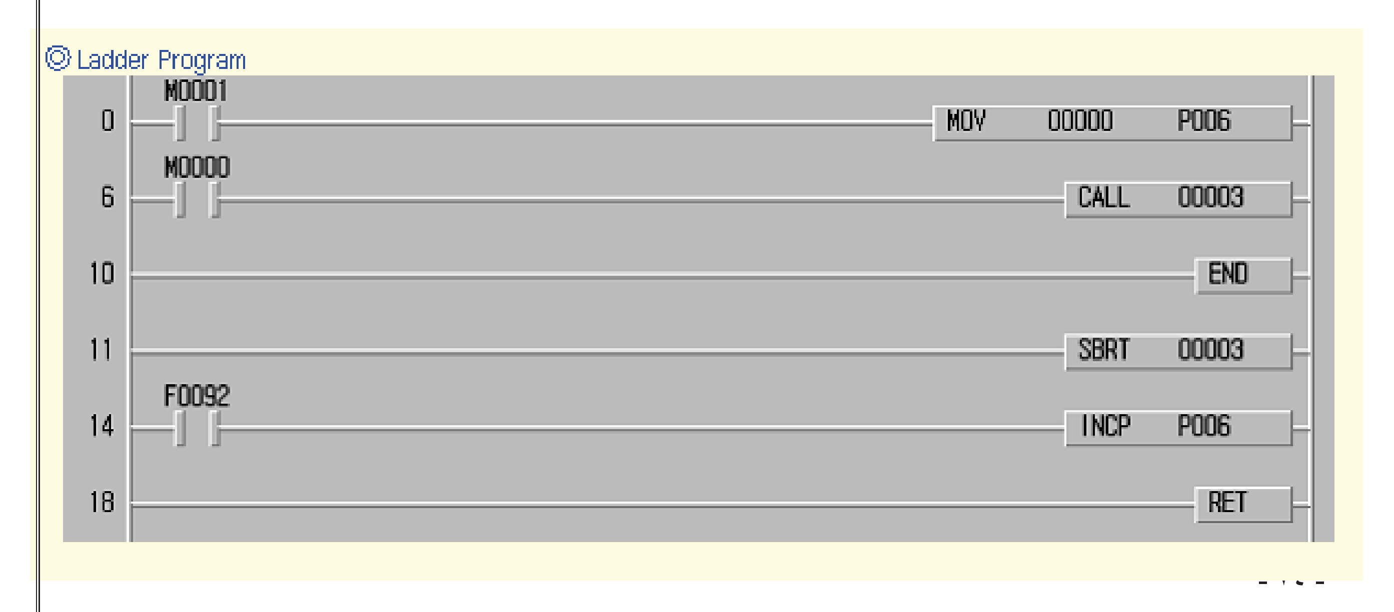
٢. Same subroutine can be called several times within the program.

٣. The position of subroutine program shall be after END Command.

٤. When n exceeds the available range or only [CALL n] or [CALLP n] Command is existed, or only [SBRT n] or [RET] Command is existed, the processing error is occurred.

٥. It is possible to call another subroutine in a subroutine max. ٦٤ times of the nested subroutine calling is available.





- **EI** ١ **Enable Interrupt**
- **DI** ١ **Disable Interrupt**
- **EI n** ١ **Enable Interrupt (not used in K٥٠٠, K١٠٠٠)**
- **DI n** ١ **Disable Interrupt (not used in K٥٠٠, K١٠٠٠)**

[EI] [EI n]

١. Makes TDI (Time Driven Interrupt) and PDI (Process Driven Interrupt) operation possible .

٢. Interrupt program set by the parameter can be executed after this command is executed.

٣. In case of K٣, K٤, K٥ and K٤E, when n is used, the interrupt specified as n is only allowable. When n is not used, i.e., when [EI] command is used, all interrupts set in the parameter are enabled.

٤. When PLC mode is changed to Run mode, all interrupts are disabled. So, to use interrupt, [EI] or [EI n] command shall be executed.

[DI] [DI n]

١.Stops TDI (Time Driven Interrupt) and (PDI : Process Driven Interrupt) operation.

٢. The interrupt program can not be executed after this instruction is executed.

٣. when n is used, only the interrupt specified as n is disabled. When n is not used, i.e., when [DI] command is used, all interrupts set by the parameter are disabled.



Example

Mnemonic Program

```
00000 EI
00001 DI
00002 LOAD  M 0020
00003 EI    01
00004 LOAD  M 0021
00005 DI    01
```

Description

00000 Step enables to execute all interrupts.

00001 Step disables to execute all interrupts.

The input condition M0020 of [EI 01] gets ON, the interrupt 01 set by the parameter. Is enabled

The input condition M0020 of [DI 01] gets ON, the interrupt 01 set by the parameter. Is disabled

Ladder Program

```
0  ------------------------------------------------ [EI]
1  ------------------------------------------------ [DI]
      M0020
2  ---| |------------------------------------- [EI   01]
      M0021
4  ---| |------------------------------------- [DI   01]
```

- **TDINT** ١ **Time Driven Interrupt**
- **INT** ١ **External Interrupt**
- **IRET** ١ **Return from Interrupt**

١. Indicates the start of TDI (Time Driven Interrupt) routine.

٢. The program after the command is executed only when TDI is occurred..

٣. In case of K٥٠٠ and K١٠٠٠, the period of occurring interrupt is ranged from ٥ msec to ١٠ sec. The start of the interrupt program is indicated by TDINT and its end is indicated by IRET Command.

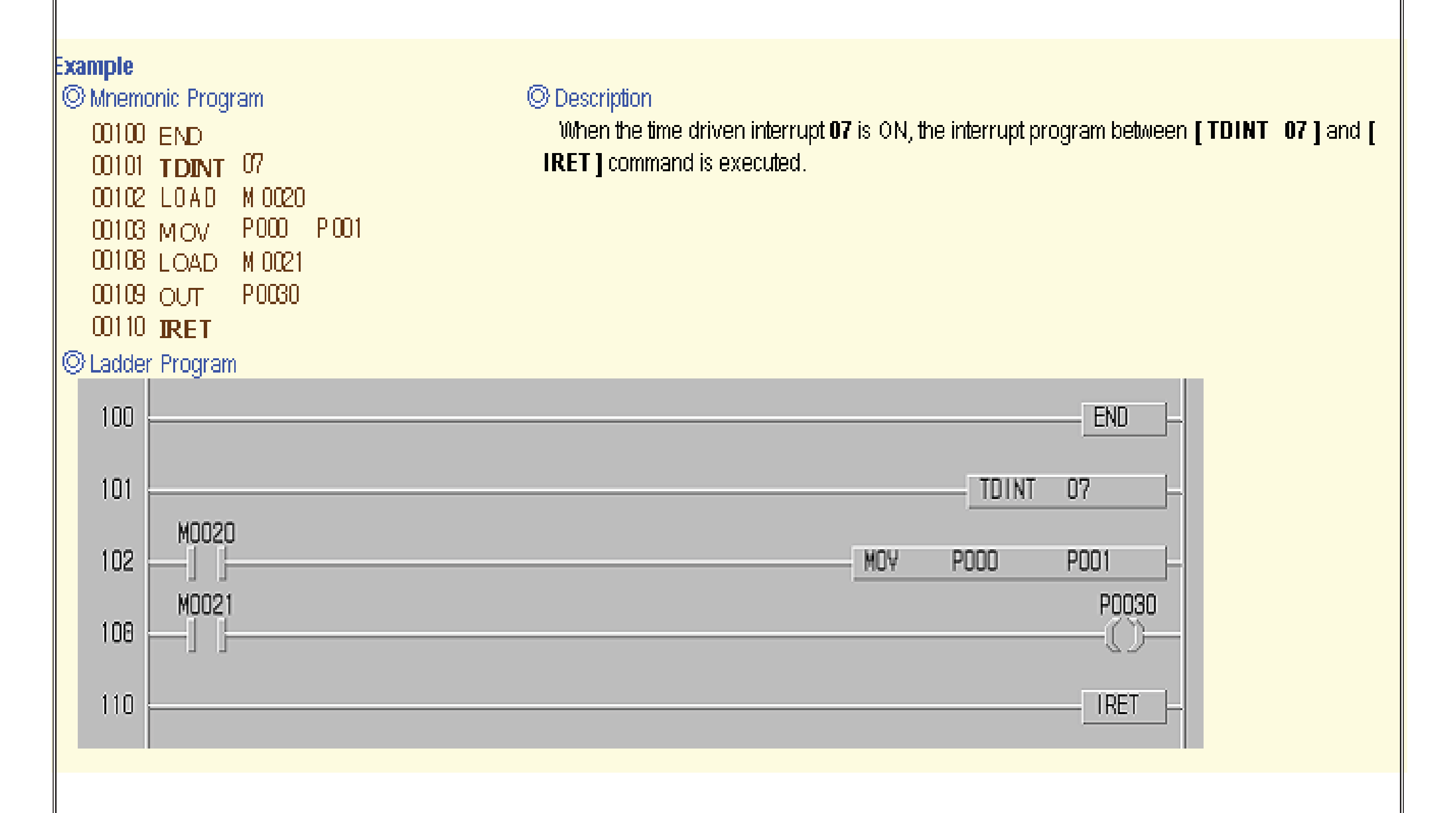
٤. In case of K٣, K٤ and K٥, the period of occurring interrupt is ranged from ١٠ m sec to ٦٠ sec. For the TDI from ٠ to ٥ or ٠ to ١٣, the start of the interrupt is indicated by [TDINT n] (n: ٠ ~ ٥ or ٠ ~ ١٣) and the end is indicated by IRET Command.

٥. The execution time of TDI program has to be set shorter than the interrupt period.

٦. TDI is not occurred during the execution of the application command.

٧. The interrupt program shall be in after END Command.

٨. To use TDI Command, the parameter, the parameter for the ‘ TDI ’ has to be set correctly.



In the following figure there are a mixer tank that have the abaility to broduce any color from the main three colour assume that pump ١ is feed the tank by the reed color , pump ٢ to feed the tank by the yellow , pump three to feed the tank by the blue color . the system have start stop bush button and have pump and valve to get the mixed color note that the system have mixer motor that operate at the mixing process . the tank take ١٠٠ second to be full of the needed color .

Note the following : -

Orange is made by ٥٠٪ reed & ٥٠ % yellow .

Green is made by ٣٠ % blue and ٧٠ % reed .

Mauve is made by ٤٠ % blue & ٦٠ % yellow .

Black is made by ٥٠ % reed & ١٠ % yellow & ٤٠ % blue .

When we bush the cancel button any process must be stop

